Dissertation

Development and Evaluation of Concepts and Tools to Reinforce Gender Equality by Engaging Female Teenagers in Coding

Bernadette Spieler, BSc MSc

Graz, 2018

Institute for Software Technology
Graz University of Technology

Supervisor/First reviewer: Univ.-Prof. Dipl-Ing. Dr.techn. Wolfgang Slany
Second reviewer: Prof. Dr. Libora Oates-Indruchová

Publication List of the Author

2018

SPIELER, B. 2018. Reinforcing Gender Equality by Analysing Female Teenagers' Performances in Coding Activities: A Lesson Learned. In *GenderIT: Gender IT,* May 14-15, 2018, Heilbronn, Germany. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3196839.3196871

2017

SPIELER, B., SCHINDLER, C., SLANY, W., MASHINSKA, O., BELTRÀN, M.E., BOULTON, H.; AND D. BROWN. 2017. Evaluation of Game Templates to support Programming Activities in Schools. In *Proceedings of the 11th European Conference on Games Based Learning*. October 5-6, 2017, Graz, Austria. p. 600-609.

SPIELER; B.; SCHINDLER, C., SLANY, W., AND MASHINSKA. 2017. App Creation in Schools for different Curricula Subjects - Lessons Learned. In *Proceedings of the 9th International Conference on Education and New Learning Technologies*. July 3-5, 2017. Barcelona, Spain, p. 5814-5824. DOI: 10.21125/edulearn.2017

AYYAL AWWAD; A.M.; SCHINDLER, C., KUMAR LUHANA, K., ALI, Z., AND SPIELER, B. 2017. Improving Pocket Paint's Usability via Material Design Compliance and Internationalization & Localization Support on Application Level. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services*. ACM, New York, NY, USA, Article 99, 8 pages. https://doi.org/10.1145/3098279.3122142

2016

BOULTON, H., SPIELER, B., PETRI, A., SLANY, W., SCHINDLER, C., AND BELTRÀN, M.E. 2016. The role of game jams in developing informal learning of computational thinking: a cross-European case study. In *Proceedings of the 8th International Conference on Education and New Learning Technologies*. Barcelona, Spain. July 4-6, 2016, p. 7034-7044. doi: 10.21125/edulearn.2016

SPIELER, B., PETRI, A., SLANY, W., SCHINDLER, C., BELTRÀN M.E., AND BOULTON, H. 2016. Pocket Code: A Mobile App for Game Jams to facilitate Classroom Learning through Game Creation. In *Proceedings of the 6th Irish Conference on game-Based Learning*. September 1-2, 2016, Dublin, Ireland, p. 61-79.

PETRI, P., SLANY, W., AND SCHINDLER, C., AND SPIELER, B. 2016. Game Design with Pocket Code: Providing a Constructionist Environment for Girls in the School Context. In *Proceedings: Constructionism in Action 2016*. February 1-5, 2016, Bangkok, Thailand, p. 109-116.

2015

PETRI, A., SCHINDLER, C., SLANY, W., AND SPIELER, B. 2015. Pocket Code Game Jams: a Constructionist Approach at Schools. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*. August 24-25, 2015. Copenhagen, Denmark, p. 1207-1211. http://dx.doi.org/10.1145/2786567.2801610

BELTRÀN, M.E., URSA, Y., PETRI, A., SCHINDLER, C., SLANY, W., SPIELER B., CABERA-UMPIERREZ, M.F., ARREDONDO, M.T., AND DE LOS RIOS, S. 2015. Inclusive gaming creation by design in formal learning environments: 'girly-girls' user group in No One Left Behind. In *Design, User Experience, and Usability*: *Users and Interactions*. Los Angeles, USA, Vol. 9187, p. 153-161. https://doi.org/10.1007/978-3-319-20898-5_15

SPIELER, B., BURGSTEINER, H., MESSER-MISAK, K., GÖDL-PURRER, B., AND SALCHINGER, B. 2015. Development and Evaluation of a web-based Application for Digital Findings and Documentation in Physiotherapy Education. In *Health Informatics meets eHealth*. *IOS Press*. Vienna, Austria, p. 182-189. doi: 10.3233/978-1-61499-524-1-182

Abstract (English)

The next generation of jobs will be characterized by an increased demand for people with computational and problem-solving skills. In Europe and especially in Austria, there is a demand for software developer experts as there is a widening skill gap in digital know-how and Information and Communication Technologies (ICT), where an increasing number of jobs remain unfilled. As a solution, the governments see a chance in supporting students early on in secondary schools and teaching them "digital skills". However, computer science topics are underrepresented in school curricula, hence teaching time for these topics is limited. From elementary through secondary school, only a few opportunities exist for young students to explore coding and today's teachers have seldom a training in computer science. Game design activities and Game Development-Based Learning (GDBL) potentially provide engaging, goal-oriented, and creative experiences in classes; in this way, GDBL is used to support the construction and transfer of knowledge in a fun and pedagogic manner. Learning theories, for example Constructionism emphasize the importance of intrinsic and extrinsic motivators in curricula, and games are a promising way to provide both while constructing the game and presenting or sharing it in public or with a community. New technologies and the emerging mobile gaming sector further the case that learning should be promoted everywhere and anytime. What seems to be a promising opportunity for all students to learn (mobile) coding in an entertaining way raises the question of whether such game based concepts also help to fix the gender gap of women in IT related fields.

Half of the working population in developed countries consists of women, but the number of women in IT disciplines is far below the number of men, especially in developed countries and across all academic levels. On the one hand, the gender disparity in technology fields is wide and deep and is a serious problem, especially at U.S., Australian, European, and Asian universities and companies that specialize in IT. On the other hand, studies show that companies with diverse employees and teams are more innovative and have a better financial performance than those that do not. Thus, to encourage more women to study these disciplines and apply for jobs in these fields is highly desirable. Women already earn about half of all university degrees, but this number include physiology or sociology, or in STEM (Science, Technology, Engineering, and Math) disciplines, biomedical engineering or architecture, which are all already more female-dominated disciplines. The real problem exists in computer science, where the percentage of female graduates has slipped every year. Middle-school girls appear to be engaged in coding courses but when they choose their future careers, only a small percentage of female teenagers plan to pursue IT as a major. Gender differences in interest, self-belonging, and self-efficacy towards ICT are already present in students aged 12 to 15 years. By summarizing all the obstacles that may seem to influence female teenagers in their career decisions,

their differences and deficits in these fields become apparent. Literature and researchers argue that the existing male stereotypes, preconceptions, the male dominated tech sector/gaming industry, gender inequality in education, the absence of female role models and mentors, and moreover, girls' presumed deficits, like low tech affinity, interests, and experiences in tech, are all reasons that lead to that gender gap in ICT. Most of these factors have caused a serious impact on the self-confidence of girls, the result being a construction of the stereotypical helpless, uninterested, and unhappy "Girl in Tech", which is even more terrifying to confirm than the male, geeky, nerdy counterpart is. Thus, female teenagers think they must be brilliant or hyper-intelligent and not motivated, interested, and focused to succeed in those fields.

To address this gender bias which is already present at an early stage, a goal of the European H2020 project "No One Left Behind" (NOLB) was to integrate Pocket Code, an app developed at Graz University of Technology (TU Graz), into different school subjects, thus making coding more accessible and attractive to (female) students. During the period of this project (2015-2017), teachers were supported to guide and assist their students in the learning processes by constructing ideas and realizing them through game design. Pocket Code helped students to take control of their own learning and to become more engaged and empowered as a result. Data were collected over a period of two years through questionnaires, interviews, focus group discussions, and data tracking within the app, as well as analysis of submitted programs according to game design and learning goal achievement. The results let the researchers conclude that the organization, and the setting of the coding courses (for example, guidance and supporting material, freedom of choice) had much more influence on female students' engagement than the game design/coding aspects or the app itself. In contrast, male students more frequently mentioned missing features in the app, and stated that learning coding was important. The program's analysis showed commonly used design patterns by genders; whether or not the learning goal had been achieved was significantly dependent on girls' different grades, group constellations, and the teaching approach used in the lesson.

After the NOLB project, the author focused on 1) the conception of a model for a more gender equal classroom setting for inclusive coding activities, and on 2) the development of a new version of Pocket Code with the aim to specifically attract female teenagers. The developed PECC model — a framework and guidelines for Playing, Engagement, Creativity, and Coding — does not only suggest an optimal starting point for (female) teenagers (e.g., to promote job clarity, make women in IT that succeed and enjoy their work more visible), but also fosters the construction of knowledge in a creative, gender-sensitive, and non-competitive environment. The PECC model has been tested and evaluated, e.g., in regard to girls' intrinsic motivation (interest, self-efficacy, sense of belonging, and fun) and the achievement of the learning goal. Although the sample size was this time very small, the numbers showed a positive trend in that girls felt more engaged. The newly Luna&Cat app integrated the results of focus group discussions and featured games that have been developed together with design students from the degree program "Industrial Design" at the University of Applied Sciences in Graz(FH Joanneum). As a practical output for this thesis, the app is available as a closed beta version on the Google Play store, and user tests in girls-only PECC environments will be conducted during the "Girls Coding Week" at TU Graz in August 2018.

Abstract (German)

Die Jobs der Zukunft sind weitreichend digital und vernetzt und die Nachfrage nach Arbeitskräften mit "Computational Thinking Skills" und Problemlösungskompetenzen steigt stetig. In Europa und speziell in Österreich besteht ein besonderer Bedarf an Softwareentwickler_innen und Expert_innen. Immer mehr Arbeitsplätze bleiben unbesetzt und die Qualifikationslücken im Bereich des digitalen Know-hows und der Informations- und Kommunikationstechnologien (IKT) vergrößern sich. Ein Lösungsvorschlag der Regierung ist, digitale Kompetenzen frühzeitig an Schüler_innen zu vermitteln. Informatik ist im Lehrplan jedoch noch immer unterrepräsentiert und die Unterrichtszeit für diese Themen ist begrenzt. Von der Grundschule bis zur Sekundarstufe gibt es nur wenige Möglichkeiten für Schüler_innen Programmieren zu erlernen. Hinzu kommt, dass das Lehrpersonal selbst nur selten eine technische Ausbildung absolviert hat. Game Design Aktivitäten und Game Development-Based Learning (GDBL) bieten interessante, zielorientierte und kreative Möglichkeiten für den Unterricht. GDBL unterstützt zum Beispiel auf diese Weise die Konstruktion und den Transfer von Wissen in einer unterhaltsamen und pädagogischen wertvollen Weise. Viele Lerntheorien, wie der Konstruktionismus betonen bereits die Bedeutung intrinsischer und extrinsischer Motivatoren in Lehrplänen, und Spiele sind ein erfolgsversprechender Weg die Motivation von Schüler_innen zu erhöhen. Spiele können so gemeinsam konstruiert, in der Öffentlichkeit präsentiert oder mit einer Community geteilt werden. Neue Technologien und der stetig wachsende mobile Gaming-Sektor fördern den Trend, Lernen überall und jederzeit zu ermöglichen. Das Entwickeln von Spielen bietet für alle Schüler_innen eine vielversprechende Chance, Programmieren auf einer unterhaltsame Weise zu lernen. Diese positive Ausgangslage wirft die Frage auf, ob solche spielbasierten Konzepte ebenso dazu beitragen, den Gender-Gap von Frauen in den Bereichen der IKT aufzuheben.

Die Hälfte der erwerbstätigen Bevölkerung in den Industrieländern sind Frauen, aber nur knapp ein Fünftel der Arbeitskräfte in der Informatik ist weiblich. Diese Geschlechterdisparität in den Technologiebereichen ist ein ernstzunehmendes Problem in technisch spezialisierten Unternehmen aber auch bereits an technischen Universitätslehrgängen. Dies ist vor allem in der USA, Australien, Europa und Asien sichtbar, obwohl Studien zeigen, dass Unternehmen mit vielfältigen Arbeitskräften, welche in Führungspositionen und Teams involviert werden, innovativer sind und bessere Geschäftsergebnisse erzielen. Daher ist es sehr erstrebenswert mehr Frauen zu ermutigen ein IT-affines Studium zu wählen und sich auf technische Stellen zu bewerben. Frauen tragen bereits etwa zur Hälfte aller Universitätsabschlüsse bei. Diese Zahl umfasst aber vor allem Abschlüsse in eher weiblich dominierten Studiengängen, wie Psychologie und Soziologie, sowie spezielle Bereiche innerhalb der MINT-Fächer (Mathematik, Informatik, Naturwissenschaften und Technik), wie Biomedical Engineering oder Architektur. Das eigentliche Problem liegt im Bereich der Informatik wo der Anteil der weiblichen An-

fängerinnen und Absolventinnen jedes Jahr sinkt. Während sich Mädchen in Unterstufe noch gerne mit Programmieraufgaben beschäftigen, sind es später nur noch wenige, die Informatik als Studienrichtung und Basis für ihre weitere Karriere wählen. Geschlechterspezifische Unterschiede in Bezug auf Interesse, Zugehörigkeitsgefühl und Selbstwirksamkeit sind bereits bei Schüler_innen im Alter von 12 bis 15 Jahren in Informatik sichtbar. Eine Analyse aller Schwierigkeiten mit denen Teenagerinnen konfrontiert sein können, verdeutlichen diese Probleme. Die Literatur und Forscher_innen argumentieren, dass folgende Faktoren junge Frauen in ihrer Berufswahl beeinflussen: männliche Stereotypen, Vorurteile gegenüber dem Berufsfeld Informatik, dem männlich dominierten Technologie-Sektor beziehungsweise die Gaming-Industrie, eine Ungleichbehandlung in der Bildung, das Fehlen von weiblichen Vorbildern und Mentorinnen, sowie angenommene Defizite von Mädchen, wie eine geringere technische Affinität oder weniger Erfahrung und ein Desinteresse für technische Bereiche. All diese Faktoren beeinträchtigen das Selbstvertrauen der Mädchen stark. Das Ergebnis ist eine Konstruktion des stereotypischen, hilflosen, desinteressierten und unglücklichen "Mädchens in der Technik". Diesen Stereotypen gerecht zu werden scheint noch beängstigender als das männliche “nerdige” Gegenstück. Viele Teenagerinnen glauben, dass sie hyperintelligent sein müssen, um in der Informatik erfolgreich zu sein und nicht, dass eher Eigenschaften wie Motivation, Interesse und Fokussierung auch zielführend sein können.

Um diesen bereits früh manifestierten Stereotypen in der Informatik entgegenzuwirken, wurde das EU H2020 Projekt “No One Left Behind” (NOLB) ins Leben gerufen. Ziel dieses Projekts war es, das Lerntool Pocket Code, eine an der TU Graz entwickelte App, in verschiedene Schulfächer zu integrieren und so Programmieren allen Schüler_innen zugänglich zu machen. Lehrende wurden dabei unterstützt, dass Schüler_innen Ideen entwickeln und selbst Spiele verwirklichen konnten. Pocket Code half Schüler_innen dabei ihren Lernprozess selbst zu steuern, sie besser in den Unterricht einzubinden und sie somit zu stärken. Daten wurden über einen Zeitraum von zwei Jahren (2015-2017) durch Fragebögen, Interviews, Fokusgruppendiskussionen und das Tracking von Events innerhalb der App erhoben. Des Weiteren wurde eine Analyse der eingereichten Programme nach ihrem Spieldesign und der Lernzielerreichung durchgeführt (das Lernziel wurde individuell von den jeweiligen Lehrpersonen definiert). Die Daten lassen darauf schließen, dass die Organisation und Durchführung von Programmierkursen (zum Beispiel Anwesenheit von Mentor_innen, Hilfe durch unterstützendes Material, Freiheiten in der Spielentwicklung, etc.) viel mehr Einfluss vor allem auf das Engagement der Schülerinnen hatten, als die Programmierung oder die verwendete Programmiersoftware selbst. Im Gegensatz dazu gaben männliche Schüler häufiger fehlende Funktionen in der App an und wiesen darauf hin, dass das Lernen von Programmieren wichtig für sie sei. Die Analyse der Programme zeigte unterschiedliche Designmuster von Schüler und Schülerinnen, aber keine Unterschiede im Programmaufbau. Des Weiteren hängt eine erfolgreiche Lernzielerreichung bei Schülerinnen signifikant von den Schulstufen in denen Pocket Code verwendet wurde, der Gruppenkonstellation und dem verwendeten Unterrichtskonzept (Templates vs. Learning-by-doing) ab.

Nach der Umsetzung des NOLB-Projektes und der Analyse der daraus gewonnen Erkenntnisse, konzentrierte sich die Autorin erstens auf die Konzeption eines Modells für eine geschlechtergerechtere Unterrichtsumgebung für Progammieraktivitäten und zweitens auf die Entwicklung einer eigenen Version von Pocket Code, welche speziell Teenagerinnen ansprechen soll. Das entwickelte PECC-Modell — ein Framework und Guidelines für Playing, Engagement, Creativity und Coding — beinhaltet zum Beispiel die Schaffung eines gemeinsamen Ausgangspunktes für alle Schüler_innen (z.B. IT Jobs definieren, Frauen aufzuzeigen, die in der Informatik erfolgreich sind und ihre Arbeit sichtbarer zu machen) und fördert auch die Konstruktion von Wissen in einem kreativen, geschlechtersensiblen und nicht wettbewerbsorientierten Umfeld. Das PECC-Modell wurde getestet und bewertet,

zum Beispiel der Einfluss auf extrinsische und intrinsische Motivatoren (Interesse, Selbstwirksamkeit, Zugehörigkeitsgefühl und Spaß) und auf die Erreichung des Lernzieles. Die Auswertung der Ergebnisse zeigen einen positiven Trend und ein gesteigertes Engagement von Mädchen unter Verwendung eines PECC Ansatzes. Die neu entwickelte Pocket Code Variante Luna&Cat beinhaltet die Ergebnisse aus dem Fokusgruppendiskussionen und zeigt als Beispielprojekte Spiele, die gemeinsam mit der Fokusgruppe und Studierenden aus dem Studiengang "Industrial Design" (Fachhochschule Joanneum) entwickelt wurden. Als praktisches Ergebnis dieser Arbeit steht die App als "closed" Beta-Version im Google Play Store zur Verfügung. In weiterer Folge ist bereits im August 2018 geplant, erste Tests mit der App in einer "Girls-only" PECC-Umgebung im Zuge einer außerschulische "Girls Coding Week" durchzuführen.

Acknowledgements

This thesis would not have been possible without the support and help of many people. First and foremost, I would like to give thanks to my husband, Patrick Spieler, who was a great source of support during this whole time and has driven me to finish this work in time.

I would like to thank my supervisor and mentor, Wolfgang Slany, for his great support and trust not only during this thesis but also during my professional career. He provided a motivational, inspiring work environment and enabled the realization of many exciting ideas and projects. I would also like to thank my external reviewer, Libora Oates-Indruchová, for her professional input and suggestions. Thanks to the European Commission, who partially funded this work by the EC H2020 Innovation Action NOLB (Grant Agreement No.645215). Furthermore, I would like to thank the staff of the Institute of Software Technology, especially Petra Pichler, and Elisabeth Orthofer for their outstanding guidance in almost all matters.

Many thanks are owed to all my colleagues and experts for their valuable input, help, and teamwork during the NOLB project: Anja Petri, Christian Schindler, Peter Kapfer, Olena Maskina, Matthias Müller, and Johanna Stefanzl. A big thanks as well goes to the whole NOLB consortium from Spain (INMARK Europe, UPM, ZED worldwide), UK (GameCity, Nottingham Trent University), and Germany (HdM/Hochschule der Medien). Special thanks to Xenia Beltrán and Yolanda Ursa from INMARK Europe, who managed the whole project perfectly. A big thanks to the teachers from our pilot schools at Akademisches Gymnasium (Ursula Sturz, Magdalena Mader), GIBS (Heinz Knasar, Patricia Raboso, Shannon Wardell), and Borg Birkfeld (Ines Voraber, Uwe Kondert). Without their support, the project and my work would not have been feasible. Thanks for providing the opportunity to work with students and thanks to TU Graz students who helped during the lessons: Elisabeth Heschl, Mario Pagger, Robert Riedl, and Daniel Neuhold. Thanks to the whole Catrobat team for their continuous feature development during NOLB, especially Thomas Schranz, Christoph Heidenreich, Thomas Hirsch, Stefan Jaindl, Florian Weissensteiner, Sebastian Gabl, Sebastian Schrimpf, Thomas Mauerhofer, Wolfgang Wintersteller, Andrej Knaus, Eric Gergeley, and Emanuel Moser. A special thanks to Amra Dzombic, who created the logo for Luna&Cat and graphics for posters and the media library. I also want to thank Mario Kolli and his students (FH Joanneum) who created the first versions of the featured games.

Last but not least, thanks to all my friends, who were never tired of listening to me talking about my work: Petra Portenschlager, Stefanie Leersch, Stefan Bacher, Sigrid Prenner, and Laurie Gugliemino. Most of all, I would like to thank my family. Words cannot describe my gratitude.

Bernadette Spieler, Graz 2018

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Graz, ______________________ ______________________

Place, Date Signature

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Graz, am ______________________ ______________________

Ort, Datum Unterschrift

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> "One thing I always tell young girls: Never let anybody tell you you can't do it. Growing up, they'd look at me like, Really? Even when I did my college visit, I had someone tell me most people change their minds after the first year. I never gave up. Even when I was having teachers tell me, just take a break from math, you can take this class next year. I said, 'No, I'm going to take it now.' I kept pushing for it."
>
> Michelle Haupt, Operations Engineer at NASA

The percentages of female university students in ICT (Information and Communication Technology) fields in Austria currently varies between 4% to 20%, with percentages dropping faster for female students over the number of their study years compared to male students, and also cumulatively have dropped more for female computer science students over the last 30 years (Binder et al., 2017; Schipfer, 2005). In contrast, job prospects in the ICT sector are great and will most likely become even better in the future (Cuff, 2015; Sangrà and Gonzràlez-Sanmamed, 2010; James et al., 2013). Unfortunately, there is also an increasingly unsatisfiable demand for ICT personnel from the industry, in particular for software developers (World Economic Forum, 2016). Over the next few years, students born during the years with low birth rates of 1990-2010 will graduate from high school, thus entering their professional careers or universities. Those will not replace the people retiring (PopulationPyramid.Net, 2018). At the same time, the demand for IT professionals will be increasing with a dramatic impact on the industrial and economic world. To counteract this, it is necessary to inspire young people, especially young women, and make IT more accessible and attractive for them.

The number of women in technical fields is far below the average number of males, especially in developed countries (Lamborelle and Fernandez, 2016; IT Manager Daily, 2018; NCWIT, 2015). Gender differences in STEM are already present in secondary schools in students aged between 12 to 15 years (Sadler et al., 2012; Tsan et al., 2016; RTE, 2016; Cukier et al., 2002; Gabay-Egozi et al., 2015; Khan and Luxton-Reilly, 2016; Zagami et al., 2015; Beyer et al., 2003; Unfried et al., 2015; Mann and Diprete, 2013; Ko and Davis, 2017). Adolescence is a critical time for identity formation, and self-attributes are a source for internal conflicts, especially for female teenagers (Charles and Bradley, 2009; Vervecken et al., 2013; Vervecken and Hannover, 2015; Carter, 2006; Appianing and Eck, 2015). It is during this intermediate female adolescence that girls begin to make critical career choices, which therefore makes this a key age to reinforce them and reduce the gender disparities in ICT. Their feelings of belonging and self-efficacy can be particularly influential in girls' interests and motivations (Stout and Camp, 2014; Baumeister and Leary, 1995; Walton and Cohen, 2007; Veilleux et al., 2013; Master et al., 2016). Therefore, the challenge is to close the gender gap that already can be seen in students of lower grades. Although significant efforts in getting more women in the IT sector are underway, the number of women who are enrolling in higher education is still very low and they are still underrepresented.

Acquiring computational thinking skills (Wing, 2006; Kahn, 2017; Tedre and Denning, 2016; Mannila et al., 2014), particularly coding, is of great importance to building a positive economic, developmental, and innovative future. Following a constructionist approach (Papert, 1985, 1993; Papert and Harel, 1991; Xinogalos et al., 2006), computational thinking skills also constitute an important part of general knowledge for all human beings from a philosophical point of view, since they allow us to understand the foundations of rational thought in a clear, easily understandable, but also inspiring and challenging way. Society and government should thus aim at allowing the female half of the adolescents to equally acquire these skills. The author believes that such goals can contribute in a meaningful, effective, and long-term way to this worthwhile endeavor.

To address this gender bias, one of the goals of the European H2020 project No One Left Behind (NOLB) included integrating the educational tool Pocket Code, a free open source app developed by the non-profit project Catrobat, at Graz University of Technology in Austria, into different school lessons (Spieler et al., 2017). Through games, Pocket Code allows teenage girls to incorporate diversity and inclusiveness, as well as the ability to reflect their cultural identity, their emotions, their likes, and their ways of interacting and thinking. Through sharing those games — in and out of the classroom — the gender biases are not simply changing but being broken down in response to the variety and diversity of girls. With this approach, Pocket Code supports participative user experience design and coding.

To evaluate the impact of the use of the app in these courses, results were captured on engaging girls in design and coding activities and a plan was formed on how to customize the app to implement their needs efficiently. In these studies, both quantitative and qualitative methods were used. Three types of data have been collected: 1) surveys conducted at different times during the cycles, 2) focus group discussions, interviews and on-site observations, and 3) analysis of created programs (game design elements, data tracking during coding resulting in behavioral analysis). Since the research area of the NOLB project was very broad and multi-disciplinary, the research scope of this thesis has to be narrowed down to tools developed and tested in Austria, as well to the Austrian pilot study and the related research questions. Therefore, this thesis primarily focuses on accomplishments related to female students.

After the NOLB project, the author continued working on the Catrobat project with schools, and especially with female teenagers, to meet the goal of exploring new ways to optimize the coding activities and the app Pocket Code for girls. The author assumed that it is possible to spark girls' attention in coding by getting them engaged in computational thinking through collaborative, creative, and engaging coding activities. By teaching young women the fundamental principles of coding, showing them a realistic picture of the people and the work in IT professions, and building a safe environment that provides room for self-expression, they get the chance to decide for themselves whether it awakens their interest. In that way coding should not be something intangible and mysterious, but a new opportunity for their lives and future career. It is essential to examine the reasons that decrease female teenagers' interest in pursuing computer science as a major in order to to mitigate these causes.

This statement reflects preconceptions from society:

> "Female teenagers do not play games, they know nothing about coding, and most of all, they have no interest in learning how to code, and do not even think of a career in IT fields."

Literature and researchers are arguing that such statements and stereotypes influence the opinions of people who work in IT fields (Cheryan et al., 2013; Bartilla and Köppe, 2016; Stout et al., 2011; Gabay-Egozi et al., 2015; Vervecken and Hannover, 2012; Master et al., 2016; Cheryan et al., 2015, 2009; Matlin, 1999). However, there also exists a stereotypical counterpart for girls and women in IT which is pictured in the summary of the literature review chapter 2.21. Furthermore, it is still true that young women who decide to enter the computer science"pipeline" are still pioneers and token women who have to fight against prejudices. The world would be a much nicer place if women in tech would be seen as role models, mentors, or simply experts in their field. Therefore, the goal of this doctoral thesis was first to examine possible factors that discourage female students in coding classes, to further analyze suitable learning conditions, and finally, to investigate girls' coding and playing behavior. As a result, the author developed first a new framework, called PECC — a model for Playing, Engagement, Creativity, and Coding, — and thus created guidelines for gender unbiased coding workshops. The related guidelines should help to increase girls' expectations for success in coding and their value in their programs, as well as the extent of the support they receive. By considering the diversity of users and their different levels of experience, the author aims at mitigating built-in assumptions about young female developers as well teachers and facilitators so they can create more inclusive and broadly appealing services and courses. Furthermore, the PECC model has been evaluated to show how it can be incorporated in a computer science lesson with the purpose of measuring girls' intrinsic motivation for such courses in terms of interest, self-efficacy, sense of belonging, fun, and also performance (learning goal achievement). Third, a new flavored version of Pocket Code has been developed which specifically targets our female teenager community; in regard to PECC, it does so through playing games and game design, engagement through gender sensibility and awareness, creativity with allowing personalization and customization, and coding by problem-based and interdisciplinary project work to foster self-directed learning and self-expression.

The findings of the thesis suggest the best situations in which girls enjoy coding and are intrinsically motivated and interested and thus, either tend to reach the learning goal more properly or have more positive feelings and realistic perceptions toward IT. This thesis should show the ways in which computer science education can make advances in addressing this gender gap in the future.

1.1. Motivation

The motivation to write this doctoral thesis resulted from the author's idea to provide better experiences for female students in coding activities. Studying computer science was also not the authors' first choice, but the subject was inherently connected with many positive feelings and the author's self-efficacy.

A number of girls-only initiatives as well as clubs, courses, and workshops for girls aim to encourage young females to get involved in coding by applying a collaborative, creative, and playful gaming approach (Paderewski et al., 2015; Hulsey et al., 2014; McLean and Harlow, 2017; Denner et al., 2005; El-Nasr et al., 2007; Subsol, 2005). Many different methods and frameworks with different characteristics and goals are meant for different levels of education or target different age groups of female students (Wu and Wang, 2012; Kangas, 2010). Various conferences and workshops have been organized around the topic of game-based learning and gamification (e.g., ECGBL[1], GWC[2], or serious play events[3]) but many of them miss the opportunity to set a focus on gender or diversity in game creation or requirements for a suitable classroom setting. For example, gender is often only a small conference track or not even a track or topic at all in such communities. Thus, they miss the opportunity to include more gender sensitivity and a diverse target group in these research areas. Although a great deal of literature in areas of games and learning exists, only a small share includes gender and diversity, and further, only a few conferences (e.g., Gender&IT[4]) or journals (e.g., Gender and Education[5]) exist which focus on this issue in particular. Many publications about girls and tech present local projects (e.g., school projects, girls-only initiatives) and most try to explain the gender gap in IT by presenting numbers or reasons through interviews and questionnaires. Only a few focus on the concept of generating a more gender sensitive and aware setting in CS at all (Margolis et al., 1999; Bosch et al., 2014; Carter, 2006). The gender gap in IT is a worldwide phenomenon and girls-only programs performed in the past already considered most of the promising strategies to engage female teenagers in coding (Milgram, 2011; Gorriz and Medina, 2000; Giannakos et al., 2014; Zagami et al., 2015; Lewis, 2007) but provided no scale of long term effect to address the overall problem of low participation of females in computer science. The literature concluded that a range of strategies are necessary to address female underrepresentation. Seeing computer science as a male dominated culture (Gabay-Egozi et al., 2015), women in the future have to evaluate their options: they either fit into this domain, or try to reshape the domain itself (Trauth et al., 2004).

Despite this fact, hardly any studies exist which focus on providing a framework that helps all students feel engaged in coding activities with a focus on intrinsic and extrinsic motivators or gender and diversity. Furthermore, there is the need to take into account components that foster inclusion (gender-sensitivity and awareness). Warm-ups and unplugged coding activities or instructions about principles of game design/elements can create more equal starting conditions to be creative, while repetition and discussions ensure an open environment with mutual understanding and engagement in every step of the course.

[1] ECGBL: https://www.academic-conferences.org/conferences/ecgbl/

[2] GWC: http://www.gwc-conference.com/

[3] Serious play: https://seriousplayconf.com/

[4] Gender&IT: https://www.gender-wissen-informatik.com/Conference

[5] Gender and Education: http://www.genderandeducation.com/about/gender-and-education-journal-editors/

1.2. Research Objectives

Essentially, the fundamental goal of this thesis was to answer the following research questions:

Research Question 1 (RQ1):

How can we organize playful, engaging, and creative coding activities to reinforce female teenagers in computer science?

Research Question 2 (RQ2):

Do girls-only PECC activities have a positive influence in girls' performance and intrinsic motivation in regard to coding?

Research Question 3 (RQ3):

What customizations are necessary in the Pocket Code tool to foster female teenagers for PECC activities?

This thesis is organized as follows: Chapter 2 provides a literature review and is divided into four parts. First, the thesis covers Computer Science (CS) education and learning theories with a focus on the constructionist theory and trends. Second, games and learning are discussed in reference to gaming elements, game design, and game-based learning theories. Third, statistics and facts are given about women in tech, which covers all levels from the industry to university and primary and secondary education. Fourth, the diverse reasons for and solutions on how to close the gender gap are noted in the last part of this chapter. This should suggest optimal conditions from the literature on how to conduct coding workshops and how to use the app Pocket Code to spark female students' interests for IT. Subsequently, Chapter 3 presents the app and the related NOLB project. Chapter 4 presents the evaluation of NOLB activities related to Austria and female teenagers; the author provides results in Chapter 5 relating to the PECC model, its evaluation, and "Luna&Cat", a Pocket Code flavor to target female teenagers, and discusses the results at the end of this Chapter. Chapter 6 concludes this thesis and describes the author's future work. Figure 1.1 displays a picture of the thesis and the relation of the different contents.

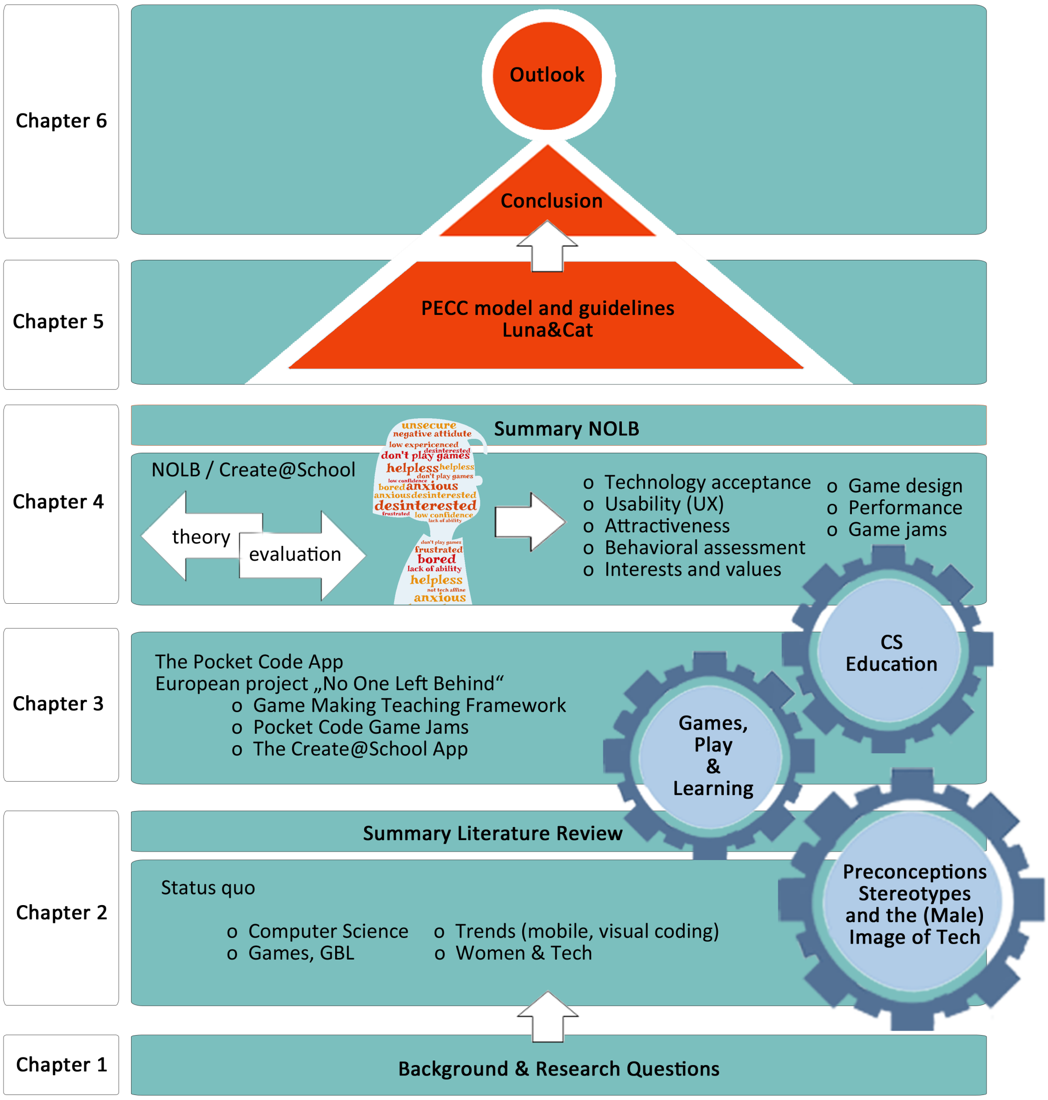


Figure 1.1.: Content of the thesis “Development and Evaluation of Concepts and Tools to Reinforce Gender Equality by Engaging Female Teenagers in Coding”

Chapter 2

Literature Review and Background

Knowledge in Computer Science (CS) is essential, and industries have increased their demand for professionals that have technical experience. The next generation of jobs will be characterized by new standards requiring employees with computational and problem solving skills in all areas, even if they are not actual technicians (Balanskat and Engelhardt, 2015). The most requested job positions are currently for software engineers, UX designers, data scientists, or analytic managers (Merrit, 2017). However, the number of young people, and women in particular, choosing to study and work in Information and Communication Technology (ICT) fields is decreasing dramatically (NCWIT, 2015; IT Manager Daily, 2018; European Commission [2], 2016). Consequently, the industry is not keeping up with the growing demand for highly skilled IT professionals (Lamborelle and Fernandez, 2016). In the last decade, European technology employment has grown three times faster than all employment in total. The continuous improvements of technology and the numerous advancements in industrial processes made it possible to develop autonomous vehicles, robotics, 3D printing, genetic diagnostics, or Internet of Things (IoT) technologies. These things are already part of everyday life, and there is a corresponding growing worldwide need for qualified scientists, engineers, and technicians of all genders. According to the European Commission, Europe may face significant shortages of up to 900,000 skilled laborers in ICT by 2020 (Lamborelle and Fernandez, 2016; Spiegel, 2010). Furthermore, it is mentioned that if there were as many women as men in the digital labor market, the EU's annual BIP could increase by 9 billion euros. Bridging the gender gap will contribute to addressing the problem, but a large-scale method to engage adolescents in ICT in a more comprehensive way is needed (Schipfer, 2005).

For these reasons, society and governments have mandated that students should acquire computing and coding skills (Balanskat and Engelhardt, 2015), or even a new way of (critical) thinking and problem solving skills (Wing, 2006; Kahn, 2017; Tedre and Denning, 2016). Several countries in Europe and worldwide have integrated the ideas of teaching these summarized as Computational Thinking (CT) skills in their curricula from kindergarten to secondary education but have not provided an official and applicable solution for these issues at all (see next section). Presenting CS as a range of diverse skills which can be taught by adapting gaming concepts for academic purposes is a generally accepted and applied concept (Paderewski et al., 2015; Hulsey et al., 2014; McLean and Harlow, 2017; Denner et al., 2005; El-Nasr et al., 2007). Thus, a gamified concept should hold learners' focuses to actively participate by activating intrinsic and extrinsic motivators (Ryan and Deci, 2000).

Recent numbers gathered by the video game industry in Europe (Global Games Market Report, 2016; James et al., 2013) and selected statistics from all over the world (Entertainment Software Association, 2016; Law et al., 2017) show that playing games is a popular leisure activity for the new generation of digital natives. Games can be played everywhere, even on smartphones, tablets, and other digital devices. Moreover, the mobile game market continues to grow faster than other game industries, e.g., the number of game apps on Google Play grew by 28% in 2017 (Jingli, 2017; Takahashi, 2017). However, such concepts follow a relatively old idea, inspired by Piaget's Constructivism theory 1948 (Piaget and Inhelder, 1967), starting with first computer programming courses at MIT in 1962 (Greenberger, 1962), and refined with Papert's Constructionism concept in 1980 (Papert, 1985). Since then, different approaches were used to motivate students for CS, e.g., teachers letting students play games to improve their participation (Game-Based Learning: GBL), or students being required to modify or develop a game by themselves (Game Development-Based Learning: GDBL) (Wu and Wang, 2012). Games can be presented in playful environments to provide an authentic context in which technical subjects can be fully situated rather than just being taught.

Secondary school is the place where students must make the critical choices which decide their future careers, develop a more realistic picture of their future jobs, and assess their career-relevant abilities (Charles and Bradley, 2009). The assumptions about why female teenagers seem to be less interested in ICT have not changed significantly in the last several decades. The literature is full of findings that document women's low experience levels towards computer sciences, their negative attitudes, and their fear of failure in computer science subjects compared to their male colleagues. Thus, the research focuses on female deficits in ICT, but it is more favorable to allow a diverse range of outcomes and include the possibility of different understandings of the discipline.

To provide a clearer image of all the issues that emerged, the first part of the literature review (see Section 2.1 and Section 2.2) summarizes the whole "coding, games and learning" topic by presenting first worldwide challenges in computer science education (see Section 2.1), and second with a focus on Austria. Subsequently, learning theories from the 20$^{\text{th}}$ century will be presented, as well as the importance of intrinsic and extrinsic motivators and acquiring computational thinking skills. This section ends by describing the driving factors from the past that pushed coding activities like block based languages or mobile learning (mlearning). Furthermore, a definition for game and play will be provided (see Section 2.2) and important game design elements will be categorized. The information is then linked with popular Game-Based Learning concepts, like GDBL and game jams. The second part of the literature review (see Section 2.3 and 2.4) focuses on women and technology by presenting a complete picture of gender gaps in STEM/ICT and university studies worldwide; the review then puts the spotlight specifically on Austria and at the University of Technology in Graz (TU Graz). Furthermore, the author tries to answer why we need more women in tech. Finally, at the end of this section, the author points out the possible social reasons which have led to the exclusion of women from tech and describes important intrinsic and extrinsic motivators which seem to be a promising direction in closing the gender gap in ICT.

2.1. Computer Science in Education

Learning new concepts, like coding or a new language, is difficult because they are complex activities which require several gradual steps. The learning process must generally be guided in order not to waste time and energy, lead to misconceptions, or to learn incomplete or disorganized knowledge (Razak et al., 2011; Wu and Wang, 2012). On the one hand, for students who want to learn coding on their own, online courses/trainings, eLearning, or MOOCs are the perfect way. These methods are also attractive in the areas of Distance Education, or Lifelong Learning. These courses reach a broad mass of people, but are often criticized, e.g., that they are using unprofessional teaching methods (Online Course Report, 2016), or that this concept does not work at all: The completion rate for online courses is only 5-10% (Frenkel, 2018). Some opportunities to learn coding also exist outside of school, e.g., attending coding clubs or workshops, or trying out block-based visual oriented programming languages for novices: for instance, Scratch, Pocket Code, Snap, or App Inventor (see next section).

On the other hand, most of the students do not feel intrinsically motivated to learn coding and will not join any off-campus activities voluntarily. Thus, for them the only opportunity exists during the regular computer science classes to gain knowledge in coding and to become more interested in tech. A German study (Milberg and Fuchs, 2009) shows the necessity for early contact with ICT topics to increase interest in such topics. The literature argues that students often find coding activities in schools difficult or boring and end up memorizing the processes without understanding them (Khaleel et al., 2015). Many concepts in programming are hard to understand for students in primary and secondary education, either due to the degree of complexity, or the level of abstraction of such concepts. Thus, the literature suggested that an enjoyable approach must be adopted in learning, especially for difficult subjects (Robins et al., 2003). Game elements and an additional fun factor influence the general outcome of the course and make it an interesting experience for all students.

The education sector in Europe is facing many challenges and limitations, but also possibilities on how to attract, motivate and engage students with content from an academic curriculum. At the same time, it is supporting the formal learning process by providing a learning experience that matches the dynamics of the 21st century (Informatics Europe/ACM Europe, 2016). More than ever, Albert Einstein's words are a reality: "It is the supreme art of the teacher to awaken joy in creative expression and knowledge." Although many pan-European initiatives try to shape school curricula more to computer science and coding, students often leave school with a lack of computer science education (Noor-Ul-Amin, 2013). The use of ICT in education requires a more student-centered learning setting to motivate and to strive for a deeper understanding through practical applications.

In addition, using games in formal learning situations is an important topic of current research but the investigation of effects of students learning in ICT are still largely underexplored. With an increased emphasis on computational thinking in curricula across Europe, schools, teachers, and the government are identifying different learning experiences for learners, which will engage them in ICT. Although many countries regard understanding the importance of ICT and teach the basic skills and concepts of computing, many fail to provide connections to real world problems or practical examples, or to teach specific digital competencies, like programming or abstraction (Bender et al., 2016). However, to adopt new pedagogical practices and integrate ICT in a meaningful way, a more uniform regulation is necessary to not only support students in private schools or with off-campus coding courses, but to provide computer science for all students to optimally prime them for the digital future. There is already a small but consistent change in the educational system and in teachers' views regarding the setup of computer science courses (see the next sections).

2.1.1. Learning ICT: a worldwide challenge.

The European Commission states, "All of Europe's citizens need to be educated in both digital literacy and informatics" (Informatics Europe/ACM Europe, 2016; European Commission [1], 2016).

Thus, IT education must be seen as an interdisciplinary field that bridges the gap between the use of digital media and information-processing technology as well as basic concepts and fundamental ideas of computer science. However, in Austria and in many other European countries, computer science topics are underrepresented in school curricula, hence, teaching time for these topics is limited (Xinogalos et al., 2006). From primary through secondary school, only a few opportunities exist for young students to explore coding and CS topics. Furthermore, today's teachers are rarely trained in computer science, which impairs their potential to motivate students in these courses.

Most of the European countries have a very similar situation[6] (Grandl and Ebner, 2017) (see an overview in Figure 2.1). In German-speaking countries or Slovakia and Poland, the focus of ICT education lies especially on the three pillars of "Computer Science", "Digital Literacy", and "Media Education" (Vuorikari et al., 2016; Hasler, 2013; Kabatova et al., 2016). England, a leader by example, introduced the subject "Computing" in 2014 (Peyton-Jones, 2013). In addition, there are many movements to integrate some basic informatics education in the US (Ladner and Israel, 2016; Office of Innovation and Improvement (OII), 2016; The White House, 2016) and in Australia (Australian Curriculum, Assessment and Reporting Authority (ACARA), 2017) or global extracurricular initiatives, like Code.org (Code.org, 2018). The problems towards CS in schools through the European countries are very similar: an underrepresentation of CS topics in (high) school curricula and the limited amount of time to teach CS. These factors impair the potential quality of engagement experienced by the students and teachers in these courses. The challenge for teachers all over the world is to engage students in computer science, and to present the relevance of computing in their own interests.

[6] https://learninglab.tugraz.at/informatischegrundbildung/

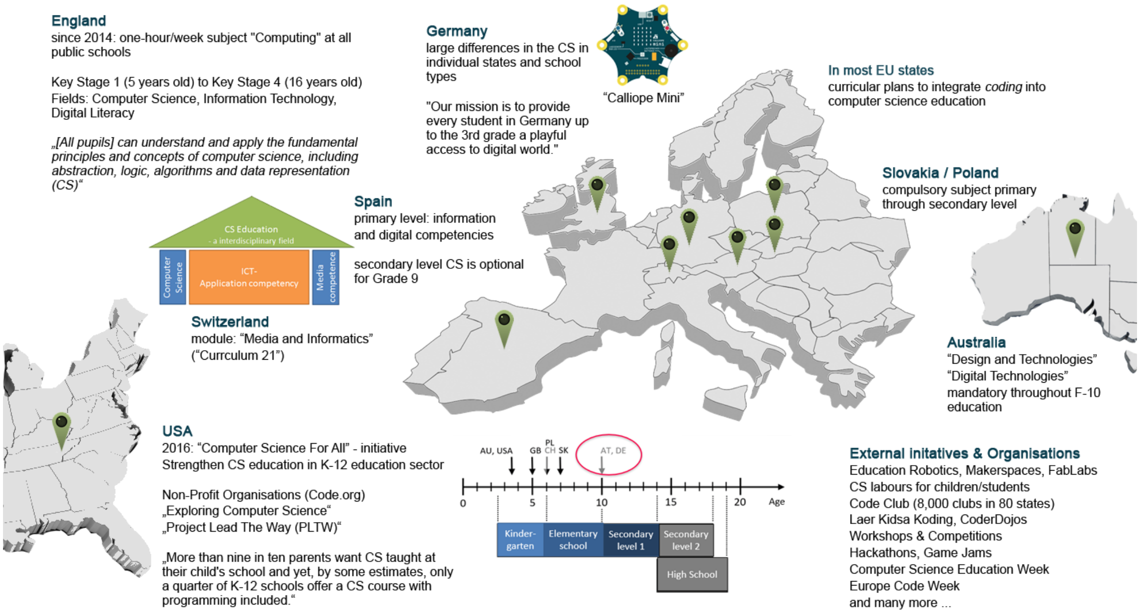


Figure 2.1.: Computer Science Education: A global comparison (Grandl and Ebner, 2017): Switzerland (Hasler, 2013; Kretzschmar, 2016), Spain (Gobierno de Espana Ministerio de Educacio, 2009), Germany/Calliope (Calliope GmbH, 2017; Starruß, 2010), UK (Peyton-Jones, 2013), Slovakia/Poland (Kabatova et al., 2016), US (Ladner and Israel, 2016; Office of Innovation and Improvement (OII), 2016; The White House, 2016), Australia (Australian Curriculum, Assessment and Reporting Authority (ACARA), 2017), Code.org (Code.org, 2018)

Efforts to redesign and rework the curriculum are obtained slowly, current solutions are often not applicable, and possibilities are limited. Thus, education programs neither offer methods or framework for CS classes nor get teachers trained specifically for this subject. Keeping up to date with changes is a challenge for all teachers. For example, in the UK, the government disestablished the Information Communications Technology curriculum, replacing it in 2014 with a new Computing Curriculum (GOV.UK Department for Education, 2013). However, the UK faces the problem of finding well-trained teachers as well, or funding the training of these concepts for their teachers (Sentence and Csizmadi, 2015; Oakman, 2016). In Switzerland, with curriculum 21, for the first time informatics contents are already integrated in elementary school (Hasler, 2013). However, the module "Media and Computer Science" does not necessarily have to be a separate subject, but integrated, e.g., in language learning or mathematics. This curriculum is regulatory from primary (5-11 years) through secondary school (11-16 years) and covers topics like design, writing and debugging programs, or understanding computer networks. In Germany, the situation is very confusing due to the large differences in CS offerings in individual states and school types (Starruß, 2010). In almost all states, in lower secondary education a "basic informatics education" is part of the curricula; depending on the state, it is integrated in a different grade during secondary level as a one unit per week. Moreover, "Calliope mini", a microcontroller should make a change and help student to have a playful access to the digital world (starting in Grade 3) (Calliope GmbH, 2017). This project, which is supported by the government, is designed to fit the European competence framework "DigComp" (a framework for developing and understanding digital competence in Europe) (Vuorikari et al., 2016). Calliope is very similar to the “BBC micro:bit”[7], which was distributed in 2016 in England for free to one million schools. In Slovakia, the subject "Informatics" ("Informatika") is available for all students starting at the age of 7 (second grade up to 11th grade), with an average of one hour per week. Here, the focus, even in the primary level, lies on computational thinking and the acquisition of programming skills (Kabatova et al., 2016). Poland included "Understanding and analysis of problems" and "Programming and problem solving by using computers and other digital devices" as a compulsory subject in its curricula, as well as "Informatics" at primary level, in middle schools, and at high schools (European Schoolnet, 2015). In general, in many Eastern European countries, computer science has the same amount of units as, e.g., in natural sciences subjects.

In the US, Barack Obama launched the "Computer Science For All" initiative in 2016, which aimed to provide computer science education in the K-12 education sector, which includes all educational institutions from kindergarten to 12th grade (Ladner and Israel, 2016). In addition, non-profit organisations like Code.org[8] have supported different projects since 2013 which foster computer science education, e.g., with age-appropriate videos, games, and courses. Code.org also promotes the “Hour of Code” event, which provides one-hour tutorials in many different languages (Code.org, 2018). Additionally, in Australia, the two compulsory subjects "Design and Technologies" and "Digital Technologies" are integrated in the F-10 education sector, i.e., from kindergarten to 10th grade (Australian Curriculum, Assessment and Reporting Authority (ACARA), 2017).

Across Europe, there are a number of extracurricular initiatives, i.e., organizations or companies that aim to bring students closer to computer science. Some examples are Educational Robotics, Makerspaces, FabLabs, such as the FabLab at Graz University of Technology (TU)[9], or student laboratories for computer science, such as "InfoSphere" of RWTH Aachen[10]. Another best practice example is the "Code Club" organization, founded 2012 in England, which works together with the Raspberry Pi Foundation. Their aim is to support teachers or other educators involved in the implementation of free "Coding Clubs" for children between 9 to 11 years old by providing materials, guidance and ideas for projects. The initiative has reached more than 8,000 clubs in over 80 countries (Raspberry Pi Foundation, 2017).

To conclude, there are many efforts in Europe and worldwide to foster students in ICT. Thus, it is important to measure their benefits as well. Figure 2.2 shows the current status of individuals with “basic” and “above basic” digital skills for each EU country (plus Norway and Iceland); the average of individuals with “basic” and “above basic” digital skills is 55% (Vuorikari et al., 2016), in Austria 65%.

[7] BBC micro:bit: http://microbit.org/
[8] Code.org: https://code.org/
[9] FablLab Graz: http://fablab.tugraz.at/
[10] School lab Aachen: https://schuelerlabor.informatik.rwth-aachen.de/

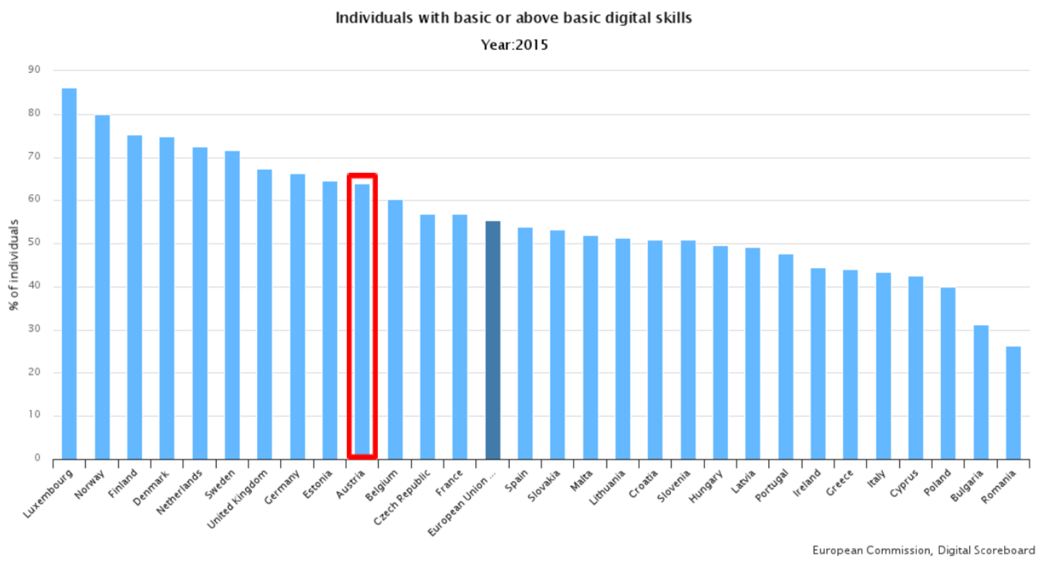


Figure 2.2.: The bar chart showing the digital skill level through Europe 2015 (Vuorikari et al., 2016).

All these actions should improve the quality and relevance of ICT training and the information flow to enable people to make better career choices and to get quality jobs and improve their lives (European Commission [1], 2016). Therefore, every student would benefit from computer literacy at an early stage, larger-scale trainings for teachers, and a standardized definition of the informatics curricula (Informatics Europe/ACM Europe, 2016).

2.1.2. Situation in Austria.

In Austria, in 2018/19, it is planned to implement the compulsory exercise Digital Basic Education (dt. "Digitale Grundbildung") in secondary schools. Thus, teachers should spend at least 2x32 annual hours by training digital content such as media competence and design, security, and computational thinking (Federal Ministry of Education Austria [5], 2017).

This obligatory regulation forces 1) schools to offer trainings and materials for their teachers, and 2) teachers to acquire knowledge to a range of new topics. Since this basic CS education should be integrated into the default curricula, teachers must find a way to apply CS to their subject. The challenge is that teachers must generally stick to an often overburdened lesson plan and have less freedom in planning their lessons depending on the subject. In mathematics, for instance, the few units available are a serious problem in Austria (Jarz, 2016). In addition, teachers can feel overwhelmed by the increasing amount of new media and different learning software products available. Interdisciplinary tools used across various disciplines and the integration of computer science (CS) with other subjects are hard to find. Most high schools are missing a higher-order thinking focus in computer science and do not teach critical thinking or problem-solving skills. Moreover, some studies indicate (Diendorfer, 2010; Dagiene and Futschek, 2008) that computer science teachers fail to provide a compelling context and motivation for those who are not already interested in these topics. Most teachers are not familiar with the field and not in a position to encourage students in an appropriate way. However, there exists an extreme pressure on schools to teach ICT, because the obligatory exercises will demand them to do so.

In Austria, computer science in high school is often an optional course and is not equally distributed over the grades. For instance, according to the Federal Ministry of Education's high school curriculum (Federal Ministry of Education Austria [1], 2017), computer science is mandatory only in ninth grade (Federal Ministry of Education Austria [2], 2017). From 10th to 12th grade it is optional (Federal Ministry of Education Austria [3], 2017) and therefore not all students are finishing school with meaningful ICT employability skills, which is a basic requirement for modern high-tech economies (Fraillon et al., 2013). The curriculum for the ninth grade (Federal Ministry of Education Austria [2], 2017) covers the basic competencies in dealing with technologies, such as Office products or descriptions of learning about basic principles of automata, algorithms, and programs, thus only low-level skills are taught. In addition, teachers are insufficiently trained for applied computer science education because it is not seen as a major subject in their education (Jarz, 2016). The reason for this is that most universities do not have programs for computer science education to prepare the future generations of teachers, or there is less interest in these programs. In the year 2016, only five students enrolled in lectureship courses for computer science at the TU Graz[11]. Moreover, for higher grades in Austrian schools, there is no officially authorized literature that contains practical, state of the art examples for computer science education (Bers et al., 2014). Even so, there are currently efforts to reinvent the computer science schoolbook with a focus on practical examples and learning materials, licensed under Creative Commons: CC BY[12].

For all these reasons, computer science should be an independent subject, which should receive an "explicit place in the context of the MINT subjects" (Hasler, 2013).

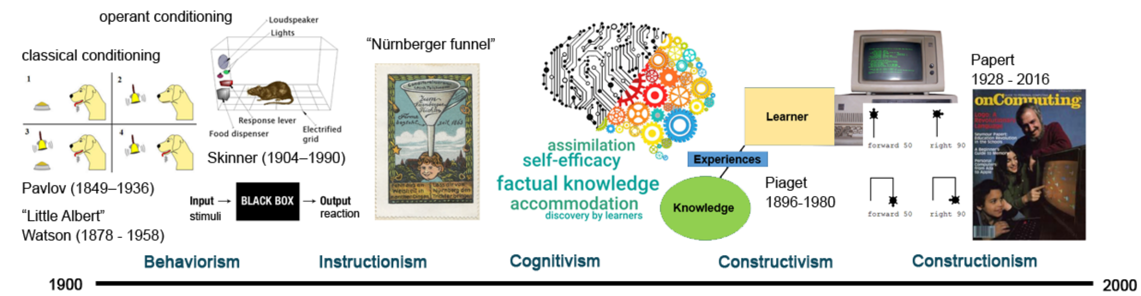
2.1.3. Learning theories of the 20th century.

Learning theories from the past serve as an organized set of principles, clarifying how people acquire, retain, and recall knowledge (Schunk, 2014; Berkley Graduate Division, 2018; Kelly, 2012). These theories helped researchers to gain a better understanding of how learning occurs and help to select appropriate techniques, tools, and strategies to support learning and teaching how to code. Three basic types of learning theories exist: Behaviorism, Cognitivism, and *Constructivism*, and some subtypes or variations, e.g., Instructionism, and Constructionism. A new learning theory from the 21st century is Connectivism, which focuses on networked learning and making connections (it uses the metaphor of a network with nodes and connections) (Kop and Hill, 2008; Downes, 2007).

Figure 2.3 provides an overview of important findings of each theory, how each theory handles students' motivations and show details on how teaching and learning should be done.

[11] Statistics students TU Graz: https://online.tugraz.at/tug$_{o}$*nline/Studierendenstatistik.html*

[12] Learning Lab TU Graz: https://learninglab.tugraz.at/informatischegrundbildung/index.php/oer-schulbuch/



	Behaviorism	Instructionism	Cognitivism	Constructivism	Constructionism
knowledge & learning	animal and human behaviors are primarily reactions to external stimuli	solely teaching and learning instructions	compromises of all processes of perception, imagination, thinking, judgment and language; builds on pre-existing cognitive structures	construct of own knowledge of the world based on individual experiences	student-centered, project-based discovery learning using new technology
motivation	extrinsic, involving positive and negative reinforcement	extrinsic	intrinsic; students set their own goals and motivate themselves to learn	intrinsic and extrinsic, e.g., learning goals or rewards	intrinsic and extrinsic *"Knowledge is not transmitted, it is constructed"*
teaching	transmitted by the teacher and absorbed by the students.	input-output-model frontal teaching "chalk and talk"	learning by providing an open environment	collaborative learning and group work	emphasises the process of knowledge construction
learning	drill / rote work repetitive practice bonus points participation points verbal reinforcement	"trivial machines"	classifying/chunking information linking concepts real world examples discussions problem solving/analogies	case studies research projects problem based learning brainstorming simulations	interacting in real-world situations building of artefacts to learn through construction, experimentation and modification, use of new media building something in public

Figure 2.3.: Learning theories of the 20th century (Schunk, 2014; Berkley Graduate Division, 2018; Kelly, 2012): Behaviourism/Instructionism (Skinner, 1976; Pavlov, 1927; von Förster et al., 2009), Cognitivism (Perry, 1999; Piaget, 1968), Constructivism (Piaget, 1968; Vygotsky, 1978), Constructionism (Papert, 1971, 1985, 1998; Papert and Harel, 1991).

To limit the scope of this thesis, this chapter focuses mostly on the theory of Constructionism and first explains the previous learning theories in short. The Constructivism and Constructionism theory was part of the author's previous work (Petri et al., 2015).

The *Behaviorism* theory researched couplings and reactions of stimuli (Schunk, 2014). Thus, several experiments with animals (e.g., mice: Skinner Box (Skinner, 1976), or Pavlov's dog (Pavlov, 1927)), and children (e.g., Little Albert (Watson and Rayner, 1920)) were performed to observe behavior (modifications) and emotions. These researchers found out how extrinsic motivators affect students' learning by first defining their initial behavior to achieve the second, desired behavior (e.g., knowledge acquisition/recalling) (Schunk, 2014). Three models were representative for *Instructionism*: "trivial machines", "black box" (von Förster et al., 2009) and the "Nürnberger funnel" (Hirschfelder, 2006). Students defined as trivial machines always respond to the same input with the same output, which is also true for black boxes. "Nürenberger funnel" became a common idiomatic expression, where the learning material is proverbially drummed into the student. The *Cognitivism* theory defined the optimization of the learning processes in all phases of life. This theory emphasized the ideas that most of human learning occurs in a social environment and that learners act according to their beliefs about their abilities and the expected outcomes of their actions, and it adopted the idea of intrinsic motivators such as, e.g., setting personal goals (Schunk, 2014). Piaget's *Constructivism* (Piaget and Inhelder, 1967) provides a framework for optimizing the learning progress at different levels of children's development (Kafai and Resnick, 1996). *Constructivism* stated that a.) teaching is always indirect (teachers take the role of a coach), b.) knowledge is shaped by experiences and built by the learner, and c.) learners were the center in the learning process. He stated that younger children create their own subjective reality, depending on their own experiences, which is suited to their current needs and possibilities. Children enhance their capability of abstract thinking and start to philosophize about probabilities, associations, and analogies by the age of 11.

2.1.3.1. Constructionism.

The *Constructionist* approach (Papert, 1985) is interested in building knowledge through active engagement and personal experience. Papert (Papert, 1985) noted that individual learning occurred more effectively when students understood the world around them and were creating something that was meaningful to them. In contrast to constructivism, the constructionist approach (Papert and Harel, 1991) was more interested in constructing personal experience and knowledge than in acquiring information. This experiential and discovery learning by challenges should inspire creativity, and project work allows for independent thinking and new ways of constructing information. The iterative process of self-directed learning underlines that humans learn most effectively when they are actively involved in the learning process and build their own structures of knowledge. In this theory, communication between students about the work, and the process of learning with peers, teachers, and collaborators, is seen an indispensable part of a students‘ learning (Papert, 1993). In constructionist environments, teachers become facilitators of collaborative groups of students who do research, construct useful concepts and increase their understanding of curriculum subjects and objectives (Clapper, 2009). Therefore, Papert implemented the concept of small, well-structured social micro-worlds, which should suit the needs and expectations of both learners and instructors as well as the affordances of technology (Papert, 1993).

In addition, Papert focused on how ideas can be formed and transformed when expressed through new technologies and different media, actualized in particular contexts, and worked out by individual minds. He figured out that students learned more efficiently if they could see a concrete result of their efforts. These can be various artifacts, such as a sand castle or a computer program (Parmaxi and Zaphiris, 2014). He argued that designing these sharable artifacts reflects students' different styles of thinking and learning. Furthermore, the constructionist approach proposed that computational ideas could serve learning in a broad variety of subjects (Tedre and Denning, 2016) and Papert's book Mindstorms (Papert, 1985) states that programming develops cognitive skills that increased the students' problem-solving abilities in many domains. Moreover, the engagement of the learner while constructing knowledge should be added with a public entity (Papert and Harel, 1991), e.g., to present in front of peers, teachers, or parents:

> "The construction of knowledge through experience and the creation of personally relevant products. The theory proposes that whatever the product, e.g. a birdhouse, computer program, or robot, the design and implementation of products are meaningful to those creating the and that learning becomes active and self-directed through the construction of artifacts." (Papert, 1971, p.2)

Thus, Papert described the huge potential of bringing new technology into the classroom (Papert, 1993). For this reason, he co-invented the LOGO programming language in the late 1960s at the MIT. Logo was designed to have a *"low threshold and no ceiling"* and was indeed used to help novice programmers, and to support complex explorations and the creation of sophisticated projects (Tinker and Papert, 1988). Logo set the basis for later visual programming tools, such as Etoys (Kay et al., 1997) or Scratch (Resnick et al., 2009; Resnick, 2017). Such block based visual oriented tools made programming accessible for a large number of people and taught new skills such as engineering, design, and coding (Blikstein and Krannich, 2013) (see next Section 2.1.4.1).

The constructionist philosophy influenced and supported the research of many, by providing a framework to integrate a playful approach in coding classrooms or courses (Ildikó Tasnádi and Farkas, 2016; Proctor and Blikstein, 2016), or by adapting it to teach new tools, like robotics (Petroviúc, 2016; Bender et al., 2016) or approaches, e.g., to use it for online learning (Cannings and Stager, 2003).

Psychologists and pedagogues from this century following the constructionist approach state three main goals. First, they wish to rethink traditional education without step-by-step guidance and to create new social and open environments (Ackermann, 2001). Second, they strive to allow students to engage in meaningful and relevant problem-solving activities, and third, they want to integrate new tools, media, and technologies in school lessons (Neo and Neo, 2009).

2.1.3.2. Intrinsic and extrinsic motivators.

One intention of using a more playful approach in classes is to motivate students. In many subjects, the students' performance and their positive grading for tests are the primary criteria (Ryan and Deci, 2000; von Glasersfeld, 1995). Students are assessed in the form of grades and reflect what they hear and read. As a result, students often learn material by rote memorization and thus, can hardly build connections due to their lack of actual understanding. There is a major difference between students only acquiring certain knowledge versus understanding and therefore applying this new knowledge. The literature distinguishes between two types of reinforcement: external and internal reinforcement. Only the external is perceived directly. Both encourage people to repeat their previous behavior.

External reinforcement provides motivation, but often the learner sets the wrong goals. Examples for external motivators are the earning of certificates or awards. Therefore, extrinsic motivation is associated with Skinner's behavioral theories of human learning (see previous Section 2.1.3). By this students are motivated to learn, but will not invest more energy in learning than necessary, thus, students, e.g., will not seek for new solutions for a given problem. However, most of the tasks given by teachers are not inherently interesting or enjoyable, thus extrinsic motivation becomes an essential strategy for successful teaching (Ryan and Deci, 2000). Extrinsic motivators are often connected with the value students assign to a task. For example, student feel more extrinsically motivated if they know it is necessary to acquire that knowledge for further job opportunities, or they are valued by others to whom they feel connected, whether that be a family member, a peer group, or a society (Brophy, 2013). Furthermore, students will more likely internalize a task if they understand it and have the relevant skills to complete it. Thus, support, feedback, and the definition of optimal challenges improves their competence.

In contrast, internal motivation builds on praising students and recognition or acknowledgements. This gives the students the feeling of pride and achievement and supports self-directed learning and practical relevance. The sense of achievement fosters a positive attitude, which further enhances their motivation. The intrinsic motivation is a more natural one, which is based on curiosity, self-efficacy, enjoyment, competence, autonomy, and interest. However, intrinsic motivation will only occur for tasks that provide intrinsic interest for an individual; such tasks have the characteristics to be novelty, challenging, or aesthetic for that individual person (Ryan and Deci, 2000). Learning theories, like cognitivism or constructivism, build on these intrinsic motivators by constructing personal experiences and promoting self-expression. In addition, intrinsic motivation results in high-quality learning and creativity. Where such support is lacking, students will feel more controlled rather than self-determined, thus their motivation will be primarily extrinsic (Brophy, 2013).

It is important to provide simulations that support different learning styles including the following: auditory, visual, and kinaesthetic, as identified by Gardner (Gardner, 1983). Thus, a teacher will not operate under the traditional concept of teaching, but rather is meant to guide and assist students in their learning processes by constructing and unpicking ideas through game making. Students are supported and encouraged to take control of their own education to become more engaged, interested, and empowered as a result (Gardner, 1991). To support teachers, a large number of different models and frameworks exist to integrate tools and concepts into curricula. One model, which is already established in many disciplines, is the Universal Design for Learning (UDL) model (Spencer, 2011). The UDL supports varied learning and assessment approaches, such as cooperative learning, performance-based assessment, or student-centered learning. Therefore, the UDL refers to the most important questions of learning: the *what*, the *how*, and the *why*. To phrase it differently: what is being taught, how the information is shared, and why the information is engaging the learners. The UDL thereby applies advances in the understanding of how the brain processes information to the design of curricula, which accommodate diverse learning needs (CAST, Inc., 2014). Therefore, the UDL provides a flexible approach that can be customized and adjusted for individual needs, and supports the creation of teaching goals, strategies, and methods, as well as materials (Brand and Dalton, 2012).

2.1.4. Computational Thinking (CT) skills.

Jeannette Wing, 2006 (Wing, 2006) shaped the term "Computational Thinking" (CT) skills. CT has been already described in author's previous paper (Spieler et al., 2017).

> "Computational thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science" (Wing, 2006, p.33)

Wing's idea that children who are introduced to CS learn more than just programming opened a new way of thinking, e.g., it showed the benefits of learning to think like a technician (Wing, 2008). Wing's findings have been incorporated into the US curriculum of many federal states of the CS (Kahn, 2017) and into K-12 movements (Mannila et al., 2014). To acquire CT Skills means to (Wing, 2006):

- Understand different aspects of a given problem,
- Link computational tools and techniques to this problem,
- Understand the limitations and power of the used tool, and to
- Generalize and apply this solution process to a variety of problems.

Wing argues that learning how to code reinforces computational thinking skills but it is not just about programming. Moreover, students should think first about possible solutions to a given problem (use of problem solving skills) and second, implement their ideas by using a computing device (use of programming skills) (Selby, 2012). In order to successfully implement their own solutions, students have to apply different programming concepts, such as loops and conditions, as well as practices, such as abstraction and debugging (Lye and Koh, 2014; Kafai and Burke, 2013). The focus on most intra- and extracurricular initiatives lies on teaching fundamental principles and concepts, problem-oriented thinking, and working in multiple ways of abstraction. As already explained in previous sections, for the teachers, applying computational thinking skills to students is actually a challenging task (Tsan et al., 2016). The literature assumes (Repenning et al., 2015; Koh et al., 2010; James et al., 2013; Kopcha et al., 2016; Syamsul and Norshuhada, 2010) that it is easier to combine concepts of algorithms and programming with activities which are engaging and fun but also relevant at the same time.

In addition, the literature and different initiatives from all over the world, point out the importance of teaching CT skills and coding to children at an early stage for several reasons. However, computational thinking is just a small subset of Seymour Papert's ideas in the 80s (see previous Section 2.1.3) and the concept of his Constructionism in 1991 (Papert and Harel, 1991). Papert was the first who used the phrase computational thinking and defined it in a much broader way. For instance, Wing focuses mainly on computer programs, whereas Papert stated that there are more kinds of constructionist projects, and computational ideas could serve learning in a broad variety of subjects, this "can change the way [children] learn everything else." (Papert, 1985, p.8)

To summarize, CT concentrates on the importance of coding and computer science activities, thus delivering concepts that are more applicable and highly essential to prepare teenagers for the future (Tedre and Denning, 2016). However, critics argue that coding should not be seen as a unitary skill but instead as a meta-skill for a complex network of other skills. Together with the 4C's for 21st Century Learning — critical thinking, communication, collaboration, and creativity — defined by the Partnership for 21st Century Learning a full range of capabilities should be taught by teachers. This is defined to be essential for preparing students for the future (Barnett et al., 2017). It is beyond the scope of this thesis to discuss in any detail the very expansive literature that exists on skills explained in different categories. Some of them will be discussed in more detail in Section 2.4.2 when explaining the importance of extrinsic factors for unbiased classrooms.

After 2006, there was a rapid increase in the number of published articles about learning and teaching CS (Wu and Wang, 2012). The current movement of promoting CS through visual programming languages has its origin at that time. The author emphasizes two trends emerged at this time, which still influence CS education today, both were important in developing the educational app Pocket Code, which plays an important part in the authors' research.

2.1.4.1. Trend 1: block-based visual oriented coding.

New technologies and tools formed the ways of learning and teaching in the 21st century. Web-based technologies like Adobe Flash and later JavaScript, CSS, and HTML5, as well as an increase in the number of modern smartphones and tablets, opened up new ways for innovative coding concepts (Kahn, 2017). In the last decade, a number of block-based visual oriented programming tools have been introduced which should help students to have an easier time when first practicing programming. These tools have all had very similar goals: they focus on younger learners, support novices in their first programming steps, can be used in informal learning situations, and provide a block-based visual oriented programming language which allows students to recognize blocks instead of recalling syntax (Tumlin, 2017). Many coding concepts that are part of the CS curricula (Goode et al., 2012) or the teaching material from Code.org (see Section 2.1.1) rely on the use of such block based visual oriented coding environments. In addition, such tools are broadly integrated in primary through secondary schools, and or even at universities, thus they have been adopted into many computing classes all over the world (Meerbaum-Salant et al., 2010). Figure 2.1.4.1 illustrates the different concepts and the most important tools for visual programming through history.

These visual and blog based programming languages originate from Seymour Papert's LOGO 1967 (Papert, 1993) which itself originates from Smalltalk/Squeak[13]. This language creates traces

[13] Squeak: http://squeak.org/

of a turtle moving across the screen allow one to draw different patterns on the screen. This teaches skills such as procedural operations, iteration, and recursion. Alan Kay, influenced by Seymour Papert and LOGO, employs the Etoys development (Kay et al., 1997; Galas and Freudenberg, 2010) which is an object-oriented application written on top of Squeak. Etoys[14] influenced the development of another Squeak-based educational programming environment: Scratch[15]. Scratch is the most popular and successful visual coding environment for block based visual oriented coding. Launched in 2007 by the MIT Media Lab, the Scratch site has grown to more than 25 million registered members with over 29 million Scratch projects shared programs[16]. Other examples are: MIT App Inventor[17], Snap![18] (Harvey and Möning, 2010), Blockly[19] (Pasternak et al., 2017), Lego Mindstorms[20], TurtleStitch[21], and GP[22]. The educational app Pocket Code[23] allows for programing games directly on the smartphone (Slany, 2014). It is based on Scratch and has been initiated and developed at TU Graz. The results of this thesis build on Pocket Code by conducting classrooms projects with the app and the app has been developed further to suit different users (school/teachers and female students). This is explained further in the next Chapter (see Chapter 3).

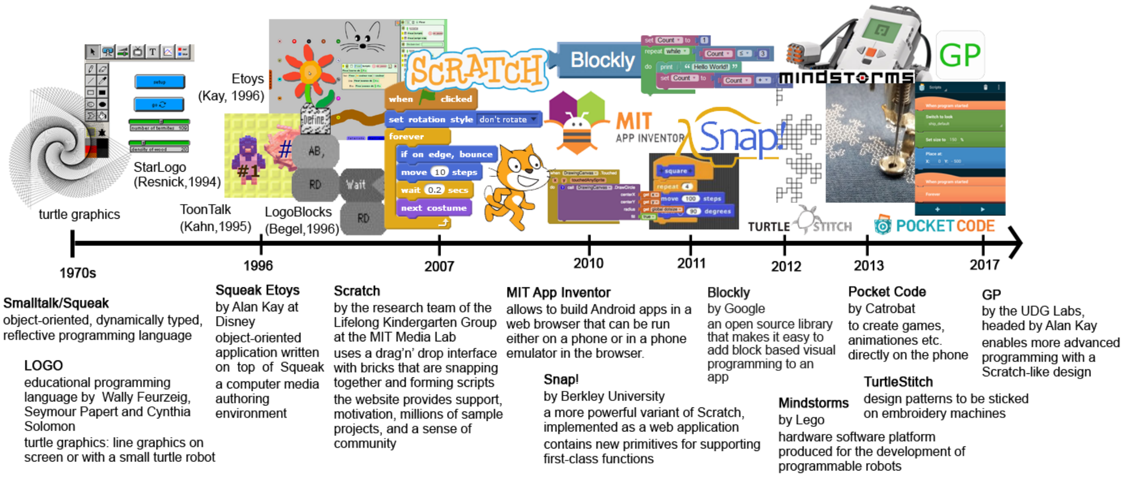


Figure 2.4.: Examples for visual/blog based programming languages: Squeak Etoys (Kay et al., 1997), Scratch (Resnick et al., 2009; Resnick, 2017), MIT App Inventor (Wolber, 2009), Snap! (Harvey and Möning, 2010), Blockly (Pasternak et al., 2017), Lego Mindstorms, TurtleStitch, Pocket Code (Slany, 2014), and GP

.

Unlike traditional programming languages, which require code statements and complex syntax rules, here graphical programming blocks are used that automatically snap together like Lego blocks

[14]Etoys: http://www.squeakland.org/
[15]Scratch MIT: https://scratch.mit.edu/
[16]Scratch MIT statistics: https://scratch.mit.edu/statistics/
[17]MIT App Inventor: http://appinventor.mit.edu/explore/
[18]Snap!: http://snap.berkeley.edu/
[19]Blockly: https://developers.google.com/blockly/
[20]Mindestorms Lego: https://www.lego.com/en-us/mindstorms/about-ev3
[21]Turtle Stitch: http://www.turtlestitch.org/
[22]GP: https://gpblocks.org/
[23]Catrobat/Pocket Code: https://www.catrobat.org/

when they make syntactical sense (Ford, 2009). Another important differentiator is the fact that all elements of the programming environment and also the programming language itself, including the formula elements, are translated to the language of the young developer. Especially for languages that are not written with the Latin alphabet, this is a huge advantage for young users, as they are not used to think in English and have very commonly difficulties to read Latin scripts. This feature is shared with Logo, Scratch, and Snap!, and certainly contributes in a major way to the very positive worldwide reception of our software. Further, in visual programming languages, a block represents a command or action and they are arranged in scripts. The composition of individual scripts equals the construction of an algorithm. The building blocks offer the possibility, e.g., to animate different objects on the stage, thus defining the behavior of the objects. In addition to the basic control structures, there are event-triggering building blocks/conditions for event-driven programming (Georgios and Kiriaki, 2009). Familiar concepts such as variables, variable lists, Boolean logic, user interface design, etc. are provided as well. Furthermore, most visual programming environments offer the possibility to integrate graphics, animations, music, and sound to create video games, animation, and interactive stories. In that way, creative and artistic talents of the students are displayed in their games, stories, and applications. Thereby, these visual languages offer the same programming logic and concepts as other (text-based) programming languages.

The previous mentioned FabLabs, Makerspaces, and Coding Clubs not only use these tools to teach programming but also refer to the whole Constructionist approach in setting up their course frameworks. Thus, concepts of “Tinkering”[24] or resources like the “Makey Makey”[25] tool became popular for activities during coding workshops with innovative forms of production and do-it-yourself work (Schön et al., 2014; García-Peñalvo et al., 2016). In addition, offline or unplugged coding activities (Brackmann et al., 2017) were often used to explain important concepts or vocabulary to students without actually using a PC, laptop, or smartphone, for explaining, e.g., x/y coordinates, the need for precise instructions for computers/robots, or variables and lists. The Internet offers many examples for unplugged coding activities, e.g., to program a classmate like a robot, paint instructions, pack a Rucksack, or send “broadcast messages” to colleagues. A list of important coding vocabulary is part of the Appendix (see Appendix A.9).

Block based visual oriented coding is becoming the standard approach when introducing students to programming. Programming is not a single skill but more a complex activity, where a student must apply cognitive skills (such as abstraction) to solve a single task (Robins et al., 2003; Or-Bach and Lavy., 2004). A lack of motivation and a missing sense of achievement can lead to negative programming experiences. In this context, researchers point out that negative educational user experiences can lead to limited participation and engagement, and thus result in negative learning outcomes (Botha et al., 2010; Laren, 2004; Romiszowski, 2004; Roschelle and Pea, 2002).

In that case, visual programming languages provide an easier start and a more engaging experience for learners. The ease of use, simplicity, and desirability of new visual coding environments enables young people to become game makers. On the one hand, researchers argue that students are not fully convinced that Scratch is a programming language which can help them learn other programming languages (Lewis et al., 2014). On the other hand, a study which compared three classes that used either block-based (Scratch), text-based (Java), or hybrid blocks/text (Snap!/JavaScript) programming languages showed that students generally found block-based programming to be easier than the text-based environments (Weintrop and Wilensky, 2015).

[24]Tinkering: https://tinkerlab.com/what-is-tinkering/

[25]Makey Makey: https://makeymakey.com/

2.1.4.2. Trend 2: the mobile way of learning.

The mobile game market is growing faster than any other game industries (Or-Bach and Lavy., 2004). The mobile games revenue in China in 2017 reached $14.6 billion, the United States market recorded $7.7 billion in revenue, and a global value of $46.1 billion was recorded, up by 12.5% each year (Or-Bach and Lavy., 2004; Piaget and Inhelder, 1967). The fastest growing regions in 2017 were the Middle East, Africa, and the Latin Americas.

Online statistics show the following (Statista, 2017):

- Leading Android gaming app genres worldwide, 2017: Casual (59.6%), Puzzle (57.29%), Arcade (55.6%), Racing (31.31%)
- The average session length spending on mobile gaming 2015 was 7.55 minutes
- Children's mobile game industry revenue in the United States in 2017 was $5.5 million
- Supercell, Electronic Arts, and Disney are among the top eight mobile game publishers
- Mobile games to reach 10 million downloads the fastest, worldwide, 2016: Super Mario Run, Pokémon Go, Clash Royale, Candy Crush Jelly Saga, and Angry Birds 2
- Users spent nearly $60 billion on apps in 2017 up ~35% from 2016 (Adams, 2018)
- The 5 best Android games worldwide, 2017: Subway Surfers, Clash Royale, My Talking Tom, Super Mario Run, Honor of Kings (Verto Analytics, 2015)
- Mobile gamers are more likely to be female, have an higher income, and are younger, compared to the online population (Verto Analytics, 2015) (more information on female mobile gamers in Section 2.4.2.4)

With mobile games, more people can engage who were previously limited to use other platforms such as PCs and consoles. Further, children nowadays grow up with mobile devices and feel comfortable using them. Considering the statistics above, current prices, and the forecast of the user penetration of smartphones in Austria, France, Germany, and the United Kingdom from 2014 to 2021 (Statista Market Analytics, 2016), as well as the difference in number of smartphone and tablet users in Western Europe in 2014 (eMarketer, 2015) and the current electronic device usage in Austria in 2016 (Google, 2016) one can conclude that smartphones will be used more by students in the future than the more expensive tablets or laptops. An Australian survey of 1,365 parents of smartphone owning children aged 3 to 17 shows that the kids spend an average of more than 21 hours per week using their devices (Telstra, 2015). Smartphones and the use of apps are already a part of our culture and are changing the way in which many people, particularly teenagers, act in social situations. For most adolescents the smartphone performs several functions of their daily lives. It helps them to organize; it has various tools, such as an alarm clock, flash, or a camera, and contributes to identity formation through self-presentation on the Internet. In addition, the smartphone is used a lot in spare time (most games are played in the evening (Verto Analytics, 2015) or for just killing some time while waiting. Online games and mobile games play an important role in the daily lives of teenagers (Bevans, 2017).

This widespread use of mobile phones is changing how learning takes place in many disciplines and contexts. Educational apps are seen as a way to encourage young people to use their mobile phones in a more meaningful way (Appolicious, 2017). Such apps are either used for providing some kind of information, e.g., "Coursera" for getting online education or "Udemy Online Courses" or "Khan Academy", or they are used for language learning, like "Duolingo" or "VoLT" (Beebom, 2017). In addition to Pocket Code, another app that help students to learn specific programming languages is SoloLearn (for the use outside of the school). Mobile Learning (mlearning) is about to be accepted by teachers in classrooms as well (Költzsch, 2017). Although the author of this thesis observed that schools are skeptical about whether smartphones should be used for teaching at all, there are many advantages to doing so. For instance, mobile phones are lighter, cheaper (Statista Market Analytics, 2016), easier to interact with, more portable, and easier for schools to maintain compared to PCs and laptops. Hence, teachers need not reserve computer labs for CS education. The option of using mobile phones also solves the problem that appropriate hardware infrastructure is often outdated and/or insufficiently available in schools. Furthermore, not all students have access to a PC for homework, but almost all of them own a smartphone. Since, modern smartphones are increasingly owned by students all over the world this could help to solve the hardware problem in and outside of schools. With the use of mobile devices, the learning process becomes more independent and flexible of the spaces previously determined for their use (OECD, 2004). During a research study (Funke et al., 2017)a requirements catalog for mlearning environments has been developed. Including but not limited to, the following technical, educational, usability, and socio-cultural requirements should be fulfilled (see full list in (Funke et al., 2017)): functionality (e.g., accuracy), security, performance (e.g., scalability, memory storage, energy consumption), pedagogical (e.g., separate views, content management, interactivity), accessibility, help-options, or other support options (e.g., customization, error tolerance, update, and configuration), communication (e.g., feedback, collaboration), and finally, portability.

To achieve long-term use in schools, it is important to evaluate and improve technology acceptance and the usability of mobile devices as learning tools among students. On the one hand, using mobile devices can make the learning environment more interactive, enhance the learning experience, and deliver knowledge in a more effective manner (Botha et al., 2010). On the other hand, most learning resources are designed for desktop and laptop applications, such as Scratch. When applying such models to mobile devices there are a few points to consider (Berri et al., 2006), e.g., the creation of pedagogical learning models to handle specific mobile learning constraints, to adapt learning expertise to suit mobile environments, and to support technically mobile learning at the network level as well as at the handheld device level. For students, a learning app must also meet the user's requirements concerning usability, ease of use, and satisfaction in order to foster the students' intent to use it (SuKuenSeong, 2016). Students need some time to realize the educational benefits of tools and thus gain the element of internal motivation later.

To conclude, mlearning methods open a new world of opportunities for teachers and students (Pereira and Rodrigues, 2013). In the future, smartphones will have faster CPUs and an increased memory and battery level. Connectivity technologies, such as Wi-Fi, 5G, GPS, and Bluetooth, allow enhancements and dynamic small-screen interface designs create the best user experiences. Teenagers increasingly have mobile devices on their own, which enables them at any time to creatively express themselves and to use apps that bring their ideas and creations to life. With a more meaningful use of mobile devices, children worldwide will acquire powerful knowledge that will make them into better problem solvers, thinkers, and learners. In addition, starting in May 2018, Samsung[26] will be promoting the initiative "mobile classroom" for the third time. A roadshow across Austria in collaboration with the programming app Pocket Code of which is very welcomed by teachers.

[26] Samsung/Coding for Kids: http://www.samsung.com/at/microsite/digitale-bildung/coding-for-kids/

2.2. Games and Learning

Much effort can be recognized from the Austrian government (Federal Ministry of Education Austria [4], 2017), the European Commission (European Commission [1], 2016), and from institutions all over the world (Code.org, 2018; Ladner and Israel, 2016) to apply gaming concepts from primary school to higher education (see Section 2.1.1). In addition, games are known as an effective approach for teachers to motivate students to interact and communicate as well as to learn (Kafai and Vasudevan, 2015). Yasmin Kafai and Seymour Papert (Kafai, 2006; Papert, 1985) point out that it is much more effective when students program games on their own instead of just learning about coding. To play an active part in the learning process and to create something meaningful allows students to collaborate and construct solutions for problems (see Section 2.1.3). Uptakes of the constructionist approach (Papert and Harel, 1991) to foster playful activities in teaching are increasing within the education communities, e.g., K-12 movements in the US (Kopcha et al., 2016). Through an interdisciplinary approach, teachers not only transfer their subject knowledge but also teach fundamental programming skills. Playful coding activities are therefore a perfect match of development of creativity, problem solving, logical thinking, system design, and collaboration skills (Backlund and Hendrix, 2013). In that way, students can learn about a specific concept or subject by developing personalized games, allowing students to freely select and use content-related preferences, like genres, themes, goals, characters, game dynamics and mechanics, backgrounds, or assets.

Before presenting the concepts of GBL, GDBL, and game jams, terms like game, play, and important gaming elements are defined within the next section as the basic background for this study. Games can be applied physically by using board games or played digitally, e.g., on consoles, PCs, tablets, or smartphones. As already mentioned in previous sections, the global video game market is increasing; in 2016, it was valued to be at 75 billion U.S dollars (music industry: $43 billion and film industry: $38 billion) (Statista, 2016). This study is focusing more on digital games and how digital gaming principles can be used in classrooms.

2.2.1. Games and play.

Games and play have always been a part of people's lives (Culin, 1975). Games were played in Egypt and Rome. The modern study of play can be traced back to Johan Huizinga's study of "Homo Ludens" (1938) (Huizinga, 2009). Huizinga describes a game as a voluntary activity bound by certain rules of time and space. In contrast to games, play is a more free activity without a set of explicit rules (Deterding et al., 2011). A game, for example, is "Hide and Seek"; it has a certain space in which the players are allowed to hide. This rule is applied at a certain time and only on the playground. People who transgress the rules have to deal with negative consequences.

The goal and also the value of a game is to be engaging, exciting, and to bring fun; and not to be a task in a traditional sense. Game value means that the game interests players to engage with the game through achievements, tasks, and appropriate goals (Shi and Shih, 2015). Besides this, a game is defined as being challenging, giving a sense of achievement, and joyful, thus it differs from the "ordinary life" (Seaborn and Fels, 2015). These factors should overlap with one's own interests. A game is a leisure activity and it is mostly driven by a goal or purpose. In many (digital) games, players solve difficult objectives, and thus they become more skilled and gain more knowledge and courage, which makes the game more challenging overall (Pereira and Rodrigues, 2013). This is defined as the "Flow of a Game" (Nakamura and Csikszentmihalyi, 2009), which builds on the interaction between challenges and skills. At the beginning, the player is usually unskilled, and therefore the game has to be simple. The player experiences playing as fun because of the novelty of the game for him or her, thus the player is in the flow. After some time, as skills improve, the player needs new challenges otherwise, the game will become boring. The more difficult the challenges are, the higher the fear of failure. Since neither boredom nor anxiety is a positive feeling, the flow state is very desirable. Flow is achieved if the skills of the player match the challenge tasks, i.e., when the right level of difficulty is chosen, see Figure 2.5.

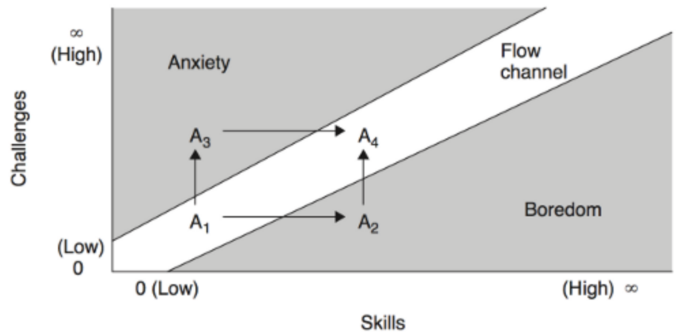


Figure 2.5.: Flow of a game (Nakamura and Csikszentmihalyi, 2009) Flow state: Challenge and pacing must rise to match skill, to support continued engagement.

In addition, Csikszentmihalyi (Nakamura and Csikszentmihalyi, 2009) presented several dimensions of the flow experience: clear goals, immediate feedback, a merge of action and awareness, a sense of control, and that the experience becomes more autotelic. To conclude, students experience flow if they do something that is challenging for them (Brophy, 2013; Chen, 2007).

The replay value is one of the critical factors of a game (Sampath, 2004). If the player is already familiar with the game, he or she knows what to expect, and therefore he or she has to be animated by new challenges through, e.g., levels, points, difficulties, or upgrades. These elements are defined in the Mechanics, Dynamics, and Aesthetics framework (MDA) which is part of the next section.

In summary, games provide a high level of engagement, through "flow" by combining elements of challenge, fun, and curiosity (da Rocha Seixas et al., 2016); thus they meet the trend of pedagogical paradigms calling for active, constructive, and playful learning (see previous sections).

2.2.1.1. Game design elements.

Most characteristics are very similar in all games. For instance, rules, goals, variables, and uncertain outcomes (Seaborn and Fels, 2015). Games are formed from a variety of components and it is the players' perception, which determines whether the experience is fun and entertaining or not. In addition, the Mechanics and Dynamics of a game motivates the players to keep on playing and to be successful (Ibáñez et al., 2014). The definition of a game supplied by video game developer and author Jane McGonigal (McGonigal, 2011).

> "When you you strip away the genre differences and the technological complexities, all games share four defining traits: a goal, rules, a feedback system, and voluntary participation." (McGonigal, 2011, p. 21)

Different dimensions exist within games that characterize them (Arseth, 2003). The game design elements can be broken into three subcategories:

- Gaming-world (e.g., level design, theme, genre)
- Game-structure (the rules of the game and the goal, MDAs)
- Game-play (e.g., the story, the player and their actions, strategies and motives)

Below, these characterizations are described in more detail. A storyboard can help to define the gameplay, the design of the game, the story, and to provide an overall picture of the whole gaming world (Co, 2006). Storyboards were used many times during the project (see next chapter) and an template is part of the Appendix (see Appendix A.5).

Gaming worlds: genre, themes, and goals.

This work was part of the European Delivery 3.1 (Martinovs et al., 2017) and part of the authors previous work (Spieler et al., 2017). At the beginning of the game design phase, elements of the gaming world can be defined, e.g., how the game is played/interactions, the scope of the game, how many levels should be integrated, and the look and feel of the whole game (Co, 2006). First selecting a genre and theme can be helpful in directing game design. Genres classify games based upon their characteristics. Video games exist largely in a commercial entertainment marketplace and have been formed around clusters (Ferreira et al., 2008; Lee et al., 2014). The classifications of game genres are unfixed and diverse. The genres in Figure 2.6 have been identified as offering useful models for gameplay for this study. These genres were used during the European project for designing new game templates (see Section 3.4.2.2), as well as for the analysis of the created games (see Section 4.4.2), and are also considered in the developed PECC model (see results Chapter 5) to support the design process during coding activities.

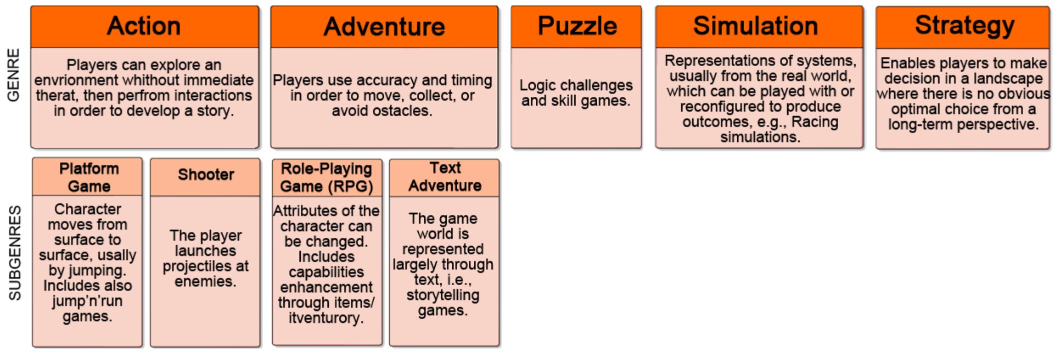


Figure 2.6.: Clustering of game genres.

Considering the diverse range of genres and their complexity in clustering, some additions to Figure 2.6 are necessary: the genre jump’n’run is a subcategory of platform games, racing, a sub-category of simulation, storytelling, a text adventure type game, and skill games, which are from the type puzzle. Often the adventure and action genre are classified as one genre because a distinction is not always clear. Other genres, which are not part of Figure 2.6, are sport, music, or educational games. In addition, two genres, mentioned in Section 2.1.4.1, are also missing in this list: casual and arcade games. Casual is a very broad term and can be defined as “a game with simple mechanics that takes skill and time to master" (Portnow, 2009). Thus a lot of mobile apps fall under this category. Arcade games are a very similar genre that uses short levels, a rapid increase in difficulty, and simple and intuitive controls (techopedia, 2018). These game genres help to define what game design elements (MDA) are necessary to effectively create the chosen genre and to help the theme 'fit' into the genre classification.

Besides the genre, other characteristics need to be defined, e.g., the players' perspective (first- or third person games), or a distinction between single- and multiplayer games (Co, 2006). In a first-person perspective, the player sees through the eyes of the main character; thus, the player is the center of action. This is very common for shooter games. The third-person view shows all the characters, as well as the interaction between the characters and the environment, which is typical for RPG games. The view has a big impact on the story of the game. On the one hand, single-player games are more structured, with rules to follow and tasks to fulfill, and on the other hand, multiplayer games rely on the development of characters, gathering experiences, and interaction with others, e.g., within the context of Massively Multiplayer Online Games (MMOG).

As for themes, no general classification exists and the theme can be almost anything. A theme can be used to describe different aspects of a game. It can refer to a specific color, story, or narrative (Brathwaite and Schreiber, 2009). For this thesis, the author will use the term "theme" to explain the gaming world, e.g., backgrounds or colors that define the game's aesthetics. Some examples for themes are criminal/detective stories, science fiction, fantasy, romance, sports, nature, future, space, realistic, horror, or comic (Co, 2006).

Whereas the theme can change during the game (e.g., with different levels), the genre usually remains the same (Rollings and Adams, 2003; Rouse, 2001). During level design, additional levels increase in their difficulty and different missions or stages could be considered, which are defined through the MDA (see next section). The level of control or interactivity of the characters does not only depend on the genre but also on the technology. Characters in PC games are commonly controlled via keys or the mouse, whereas in mobile games, finger positions or sensors are used.

Finally, the definition of the main character is important. Some games refer to the main character as an "avatar" (Brathwaite and Schreiber, 2009). Avatars are more a direct representation of the player in the game. Examples for avatars are, e.g. Lara Croft, or a shoe in Monopoly. Some game genres, e.g., puzzle or quizzes do not have a main character; they are typically necessary in action or adventure games.

Game structure: mechanics, dynamics, and aesthetics (MDA) for gameplay.

Overall, the gameplay explains the course of the game, rules of the game, and elements such as actions and levels (Co, 2006). According to Salen and Zimmerman (Salen and Zimmerman, 2003) there are three types of rules that structure games: operational (the rules of play), constitutive (underlying formal structures presented to the player), and implicit ("unwritten rules" of a game). The rules among other elements (e.g., User Interface-UI) determine the level of gameplay. At this stage, a higher-level concept can be designed that integrates characters, assets, and other elements like a status display. It is important for young game designers to know how the elements can be organized and in which situations specific elements and structures are appropriate. The key elements required for building applications are based on the Mechanics, Dynamics, and Aesthetics (MDA) Framework (Hunicke et al., 2004). Game MDAs provide a consistent structure to define game elements, goals, and rules, thereby delivering a common framework and vocabulary for games. The MDA is a formal approach to understand games and their elements in order to support the process of designing and developing a game. It formalizes the usage of games by breaking them into their distinct components of rules, systems, and “fun”, and establishing their design counterparts of Mechanics, Dynamics, and Aesthetics (Hunicke et al., 2004). The MDA framework is shown in Figure 2.7. From the designer’s perspective, the mechanics generate dynamics, which in turn leads to particular aesthetic experiences. From the user’s perspective, the process flows the opposite way. First, the player has contact with the aesthetic component of the game, which is based on the game dynamics, defined through interaction with the game mechanics. The MDAs have been also defined in the NOLB Delivery D3.3 (Martinovs et al., 2017).

In other words, rules are instantiated during gameplay, influenced by the player’s inputs, forming the dynamic of the game. The aesthetics are the emotional responses of the player. The MDA does not refer to game design beyond the gameplay, like storytelling or user experience. Below, a more detailed description is provided for the three parts of the model (Bohyun, 2015):

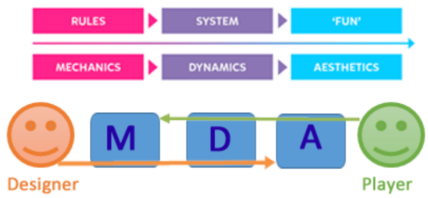


Figure 2.7.: MDA Framework (Hunicke et al., 2004).

Mechanics describe the particular components of the game and the constructs of rules or methods for the gameplay. The game mechanics include points, levels, challenges, virtual goods and leaderboards:

- **Points** are a central building block of game dynamics, and are used to reward users for the completion of activities, or accomplishing certain behaviors.
- **Levels** are the point thresholds or milestones that a player must achieve in order to be granted access to additional functionality and content.
- **Challenges/Missions** provide people with missions to accomplish and then reward them for doing so.
- **Virtual goods** are non-physical, intangible objects that are purchased for use in online communities or online games, e.g., a place to spend points.
- **Leaderboards** show users their scores and rankings in relation to others.
- **Gifting** is a powerful acquisition and retention tool.

Nah et. al. (Nah et al., 2014) listed the following as also important:

- **Notifications** are a form to provide users with feedback if a desired or undesired behavior is done. Feedback must be present either as a consequence of user action or to lead the way by guiding the user to the next objective.
- **Timers** are shown within the game and set to a specific limit, this can be categorized under the mechanics challenges.

Dynamics are defined through interactions of users with the game mechanics and describe the play of the game when the rules are set in motion. Gamification applies the game dynamics and game mechanics to get users to participate and keep them engaged. They comprise rewards, status, achievement, self-expression, competition, or altruism:

- **Reward** is the mechanism of earning points, virtual goods, levelling up, and any completing achievements.
- **Status** is defined as the rank or level of a player.
- **Achievements** are virtual or physical representations of having accomplished something and are similar to levels in that a milestone must be reached before the achievement designation is awarded.
- **Self-expression**: an individual's avatar can act as a rich focal point for expression.
- **Competition** is given if people gain satisfaction by comparing their performance to that of others. This can be achieved through, e.g., a high score display.
- **Altruism** is provided if gift-giving is a strong motivator in a community where people seek to foster relationships.

Aesthetics refer the player's experience with the game. Additionally, aesthetics describes the desirable emotional responses evoked in the player while interacting with the game system. Aesthetics are a matter of taste and it is impossible to design a game that suits everyone (Bergström et al., 2010). They are the reason for playing games and comprise, for example, fantasy, narrative, fellowship, and discovery:

- **Sensation** is when the player experiences something completely unfamiliar to them.
- **Fantasy** is where the player is caught up in an imaginary world, and tied into something that they feel could exist.
- **Narrative** creates a story which drives the player to keep coming back.
- **Challenge** is when the player feels they need to master something.
- **Fellowship** is portrayed when a community is formed that the player is actively a part of.
- **Discovery** is generated by the players need to explore.
- **Expression** is when players use their own creativity or leave their mark.
- **Submission** is when a player literally 'submits' themselves to the game.

With some exceptions, most games have a goal or common objectives which represent one core concept of game design (Shi and Shih, 2015). The designer must think about what kind of experience they want to provide to the player. The goal of the game is some kind of victory (Brathwaite and Schreiber, 2009) and sometimes referred to as missions or quests. Goals typically provide rewards, e.g., to level-up or the option to buy more advanced equipment. Games can have different kinds of goals, e.g., short, medium, and long-term goals (Swartout and van Lent, 2005). Whereas short and medium-term goals should satisfy the player, long-term goals represent guidelines for the whole game. These kinds of goals should motivate the player through the game. The following goals have been summarized (Brathwaite and Schreiber, 2009; Dormans, 2012):

- Territorial acquisition: the player occupies territory, not necessarily harming other players (e.g., The Settlers of Catan, RISK).
- Collection: collect a certain number of objects throughout the game (e.g., Pokémon).
- Solve: solve a puzzle or crime (e.g., Cluedo).

- Chase/race/escape: anything where the player is running towards or away from something (e.g., Police Chase Crackdown).
- Spatial alignment: anything that involves the positioning of elements (e.g., Tetris or Tic-Tac-Toe).
- Build: advance your characters or build your resources to a certain point (e.g., The Sims).
- Negation of another goal: the game ends if you perform an act that is forbidden by the rules (e.g., Jenga or Twister).

Finally, in the production phase, assets and sound are designed and the story is applied to the game. In the game itself, head-up-display (HUD) elements are often used (Poitschke et al., 2008). These provide information, e.g., about the status of the game. Examples include score, energy level, time, and compass or text output, thereby ensuring that the screen remains structured. Other menu screens are displayed before the game starts or are visible by pausing the game.

To summarize, typical game elements include players and their roles, objectives, procedures, rules and underlying game mechanics, resources, an underlying conflict, obstacles, and a goal.

2.2.1.2. Game design process.

To transfer the concept of games in a consistent structure, several strategies are possible. For the European project, the team used the term “ceremony” or “Shape of a Game”. In an article for EDGE Magazine (Penn, 2005), Gary Penn describes that the term “ceremonies” came from a need to classify the components of an entertainment product, and has evolved to confirm the occurrence of events for the player;

> “Ceremonies are incentives to play and sustain player interest. Ceremonies frame and punctuate play, principally the start and end of play.” (Penn, 2005, p. 92)

This shape of the interface should support overall clarity, help focus on the relevant elements of the game, and follow a clear method of interaction (Martinovs and Barrett, 2016; Martinovs et al., 2017). During gameplay, different stages, such as title screen, instructions, pause, restart, options screen, game over screen, or high score table can be used to give the player the sense of progression, along with other common concepts, such as score, energy, or timer. To use the same shape for all games during the project[27] and for the game templates (see Section 3.4.2.2), the structure has been developed as follows:

- Title screen: name/title of a game
- Instructions screen: conveys goals and rules
- One or more levels: use of the word “level” creates a connection to commercial games.
- End screen: this is linked to the end of a story, the achievement of a target/goal (game over or win screen)

Together with the described game elements of the previous sections, the shape helps the game developer to define:

[27] The project drew upon the work of partners’ visiting speaker Gary Penn, formerly Creative Director at DMA Design (maker of the original Grand Theft Auto game).

- what game design elements are necessary to effectively create the chosen genre
- what the interface is capable of
- will the theme 'fit' into the genre classification (or the genre fit into the theme)

It is useful to structure the program into sections to visually simplify and map out the structure of the program. It shows the progress through the game from the start screen; the levels and the options depend on the choices made. As the games become more complex, the structure would expand, giving place to multiple choices depending on the game mechanics and chosen game.

For game development cycles, concepts of agile and iterative software development can be used to leverage this process and to see first results very quickly (Sutherland and Schwaber, 1995; Davies and Sedley, 2009; DeMarco-Brown, 2013; MacConnell, 1998). Figure 2.8 visualizes the process of game design in reference to agile methods. As the scope of this thesis is limited, only a simplified life cycle is visualized to describe the process of how a team works in iterations to deliver or release software.

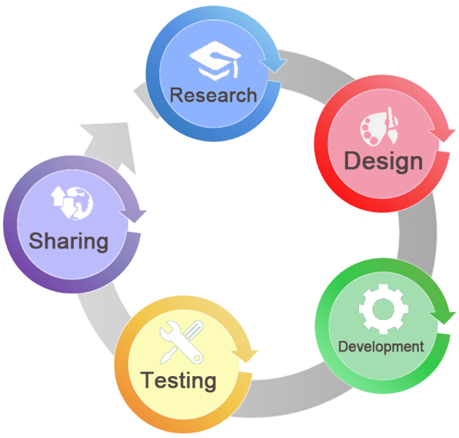


Figure 2.8.: Agile Game Development Cycle (Sutherland and Schwaber, 1995; Davies and Sedley, 2009)

The first step, *Research*, includes the development of the core idea by producing a simple game concept or a storyboard. In this phase the story, title, genre, and theme of the game should be selected, as well as a rough concept about the structure and gameplay (see previous section).

In the second step, during the *Design*, the artwork, game content and other elements (characters, assets, avatars, etc.), and the whole gaming word is produced.

The *Development* phase, includes all of the actual programming, followed by *Testing* the code and the software (playtesting). Several iterations between testing and bug fixing are possible.

As a final step, the *Sharing* phase is planned, e.g., sharing in public or with a community. In agile development this could include several beta releases or a final release for end users. The agile model required to get started with the project works to bring customer satisfaction by rapid, continuous delivery of useful software.

> "In an iterative methodology, a rough version of the game is rapidly prototyped as early in the design process as possible. This prototype has none of the aesthetic trappings of the final game, but begins to define its fundamental rules and core mechanics." (Salen and Zimmerman, 2003, p. 11)

To phrase it differently, before concentrating on every detail of the game, focus on the smallest step the games needs for playing it, e.g., limited interface control but basic game functionality. This should help novice programmers/designers to narrow down the development process.

2.2.2. Game-based approaches for learning.

As already described in previous sections, playing video games or mobile games is a popular leisure activity for young people; however, creating a game seems to be a difficult task. Engaging activities are well-received by students and contribute to improving their overall motivation and productivity (Khaleel et al., 2015). Within suitable game-based learning environments, learners are encouraged to be creative and collaborate with others. In addition, learners acquire conceptual understanding as well as practical skills (Singer and Schneider, 2012). Hence, the use of digital games as part of the formal academic curriculum comes as a natural response (Kerr, 2006). The concepts of intrinsic motivation (see Section 2.1.3.2) and the flow of experiences are best applied if students are freely engaging in self-chosen activities with a playful twist (Brophy, 2013).

Li and Watson, 2011 (Li and Watson, 2011), and Wu and Wang (Wu and Wang, 2012) divided game-based learning computer courses into categories. In addition to GBL, the game development-based learning (GDBL) approach is a very promising approach for learners to gain knowledge in coding and is aligned with the Constructionism theory through allowing users to construct their own games. GDBL refers to game-themed assignments (O'Kelly and Gibson, 2006), e.g., constructing new games, or completing or modifying existing games and the use of graphical and simplified learning tools (Cooper, 2010; Meerbaum-Salant et al., 2010). The GBL approach, in contrast, fosters learning through gameplay and Gamification, as a third category uses external motivators to repeat desirable actions of students (Ziesemer, 2013).

First, this section explains different GBL approaches, e.g., to play games, or to use Gamification elements to motivate students, and summarizes serious games as well. Second, and most important for this thesis, it describes the GDBL approach and how students can be intrinsically and extrinsically motivated by creating games. Finally, an overview about the game jam approach is provided which was applied during the NOLB project (see next Chapter).

2.2.2.1. GBL, gamification, and serious games.

Within a *game-based learning (GBL)* environment, students are able to unleash their own creativity, express themselves, and connect with their classmates, all of which promotes their social inclusion (Chandrasekaran et al., 2012; Craig et al., 2013). By adapting learning content into something that appears to be a game, a new experience for students is created. To meet the learning objectives and change them into something that seems more like a game, teachers have to adapt the context, the set of graphics assets, and the set of game mechanics. Thus, learning can perform as an active process based on the learners' interests, curiosity, and experience. Establishing this educational practice requires understanding of new learning principles and content-based curriculum issues, students' collaborative-learning processes, and the development of new concepts of projects based on learning knowledge. In groups, students are able to solve challenging problems that are open-ended, curriculum-based, and often interdisciplinary. A classroom setting that allows a hands-on approach additionally provides extrinsic motivations for students (Backlund and Hendrix, 2013; Romero, 2012). Therefore, the internet enables competition to develop across countries, and software developments enable cross-platform gaming on the same theme (Chatham et al., 2013). To encourage students to participate during the lesson through gameplay is an innovative learning approach possessing great educational value. With game challenges that encourage the player to solve problems in the game environment, a flow is provided. These challenges, e.g., can be solved through player's experiences from previous game levels (Syamsul and Norshuhada, 2010). Some advantages of games are (Lieberman, 2006): that the player has an active experience, games foster learning-by-doing, games are engaging, games promote behavioral learning, games offer consequences (direct mapping through success and failure), and games provide role models for the player. Digital game-based learning (DGBL) is related to GBL, but restricted to digital games (Kerr, 2006).

Using a *gamification approach* means to try to keep users or students motivated to perform certain tasks (Ziesemer, 2013; Glover, 2013; Sheth et al., 2012) and integrate elements, such as points, badges, and leaderboards, to maintain user activities (Deterding, 2012; Zichermann and Cunningham, 2011). Thus, games offer a reward system to externally motivate (see Section 2.1.3.2). Gamification does not always involve the use of an actual game; it often just makes use of gamified elements or patterns traditionally found in games (Hamari and Koivisto, 2013). Gamification involves the usage of game-based mechanics, aesthetics, and dynamics (see Section 2.2.1.1) to engage students, motivate them, promote learning, and help them in solving problems (Seaborn and Deborah, 2015; Gloria et al., 2014). The goal is to improve the user experience (UX) and user engagement and a deep understanding of users' needs is required. In such settings, the player performs tasks and is rewarded for them, which should increase their engagement. However, not reaching the learning goal can lead to stressful situations, which discourages the users from continuing to play (Zichermann and Cunningham, 2011). Playing is always associated with competition, trial, error, and failure, thus it is essential that the student maintains a positive relationship with their failures and learns through them (Hamari and Koivisto, 2013). The social nature of this approach allows students to engage with pre-prepared content, and game design elements can be used as a tool to control and generally increase student participation (Kapp, 2012). However, gamified learning interventions have a larger impact on students who are intrinsically motivated by playing games overall (Hamari and Koivisto, 2013).

Serious games are not simply for entertainment; they use playful and educational elements that can be reused in school curricula (Razak et al., 2011). A key factor for games is voluntariness (see Section 2.2.1), thus, in a school setting, the game has to be very challenging in order to be satisfying. A serious game (Gloria et al., 2014; Charsky, 2010) should meet learning objectives and at the same time make playing it fun, provide problems to resolve, enhance the skills level, and provide game mechanics to entertain and engage the player. Most popular types of educational games use various strategies, such as quizzes, puzzles, and problem solving activities (Shiratuddin and Zaibon, 2010). Different categories for serious games exist (Rebolledo-Mendez et al., 2009), e.g., Health games, Advergames (e.g., Second Life), Games for Training, Games for Education, Games for Science and Research, Games for Production, or Games as Work. For this thesis, Games for Education are the most important. These games have the purpose of helping their users learn something beyond entertainment, so that the player gains knowledge or skills about specific topics (Michael and Chen, 2006). Developing serious games is not an easy task because it requires a balance between theory, content, and game design elements (Ferdig and Winn, 2009). Thus, serious game developers tend to use an iterative approach of designing, prototyping, and play-testing, as well as including the target audience during this process.

2.2.2.2. Game Development-Based Learning (GDBL).

Serious games, as an overall term, involve all aspects of education, e.g., teaching, training, and informing (Michael and Chen, 2006). GBL is the practice of creating serious games to improve teaching activities and initiatives by virtue of its factors concerning engagement, motivation, role-playing, and repeatability. The goal of Gamification is to apply game elements in a non-game context. Thus, GDBL, which is defined as learning by making games, includes all these aspects (see Figure 2.2.2.2).

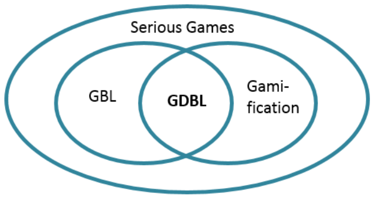


Figure 2.9.: Relationship of Serious Games, GBL, Gamification, and GDBL.

In this constructionist-learning environment, learners are encouraged to develop their own games and make design decisions (Kafai, 2006). Benefits of designing games include enhancing problem-solving skills, fostering of creativity and self-directed learning, and learning how to code in a fun way. The game construction process encourages further collaboration and engagement through teamwork within the classrooms. Therefore, creating personalized games is a popular approach to introduce novice users to programming, to conceptual thinking, engineering, art, and design.

> "In this context, not only can a game be used for learning, but also the game development tools be used for studying relevant topics within computer science, software engineering (SE), or game programming through motivating assignments." (Wu and Wang, 2012, p. 1)

Game creation challenges in schools potentially provide engaging, goal-oriented, and interactive tasks in classes, thereby supporting the transfer of knowledge in a fun and pedagogic manner (Romero, 2012). Thus, the lecture becomes more dynamic, collaborative, and attractive for students. This can be achieved by:

- Stimulating computational thinking skills, such as thinking abstractly, and the deconstruction of a problem into smaller pieces (decomposition and composition)(see Section 2.1.4).
- Moving beyond participation and creative thinking via game-making and coding (see Section 2.1.4.1).
- Using new tools and allowing students to be creative (see Section 2.1.4.2).

According to Romeike (Romeike, 2010), further aspects should be considered to promote creativity and motivation in computer science education:

- The task to be solved should be new and not offer a concrete solution in order to stimulate ideas.
- The task should be defined openly, may provide a framework, and should have an expected learning goal predefined by the teacher.
- It should be open for extension, changes, optimization, and personalization.
- Learners should be provided with basics in conceptual knowledge to allow a problem-oriented, critical, and creative thinking environment.
- To allow experimentation without a time pressure and to foster cooperation and collaboration.

The idea is to initially concentrate on designing a game and then to think about the needed programming concepts to create the game itself (Khaleel et al., 2015). Therefore, playful learning environments (PLE) are created (Kangas, 2010), which should support, e.g., playfulness, creativity (by using new technology and designing artifacts, or games), collaboration (by co-creation through the design and play processes), and emotions as a key role in thinking and learning. This research first describes the importance of an orientation lesson for teenagers to provide them some basic knowledge as well as small group work sessions to dig deeper into certain topics. Second, they emphasise to help students in creating their game ideas by designing "what if" worlds and by using templates, sharing ideas with the whole class, and planning specific elements of the game, like characters and sounds. In this research, a playing phase was also taken into account to engage in role-play and emerge appeared threads (kind of unplugged games activity). The goal was to let students make creative play worlds with narratives and plots, which resulted in a more creative, and playful learning (CPL) model which focused on the game design and creation. Thus, it supports a variety of learning outcomes: academic achievements, physical skills, co-creation, and thinking skills.

Two ways exist to integrate GDBL into school lessons; either by letting students create games by scratch or by using game development frameworks (GDFs)(Wu and Wang, 2012). In addition, many different roles can be occupied within a game development process, e.g., game programmers, game designers, animators, musicians, or play writers. Several existing game development kits (e.g., Microsoft's XNA[28]) allow students to either modify existing games or to develop their own new games with or without programming. GDF in relation to the Constructionist approach connects the educational tasks with the material artifacts and overall fits very well with the knowledge construction process. The literature review to GDBL/GDF from Wu and Wang (Wu and Wang, 2012) describes the five steps within such lectures: 1) to identify explicit course aims, 2) to exercise design and selection of GDFs, 3) to give a tutorial lecture to introduce the GDF to the students, 4) to perform a starter exercise, which should be easy and allows the students to get familiar with the development environment, and 5) to implement a game. These steps go along with collaboration (group work), and (technical) support. For GDFs it is most important for students to have sufficient documentation for their usage. It should also be easy to learn, allow rapid development, and allow flexibility and customization. The evaluation of the reviewed articles concludes that the students showed an increase in motivation, self-confidence towards CS, scores/performance, understanding, and engagement in those courses (Owston et al., 2009; Johnson et al., 2016). The study (Wu and Wang, 2012) also mentioned a number of impact factors which can influence the outcome/success of GDBL courses: teamwork (appropriate size and working environment), student's background knowledge (coding experiences), teachers' requirements (technical experiences), and time constraints and workload (limited time, task completion). To ensure that students have enough time, groups should define their own schedule, check their goals for being realistic, monitor the process, and guide the students in their learning. Figure 2.10 presents the results of this study. The model contains four elements (course aim, pedagogical theory support, GDF resource pool, impact factor), six steps A-F to help teachers or researchers to apply the GDBL design process, and two subjects (students and teachers).

[28]Microsoft XNA: https://www.microsoft.com/en-us/download/details.aspx?id=23714

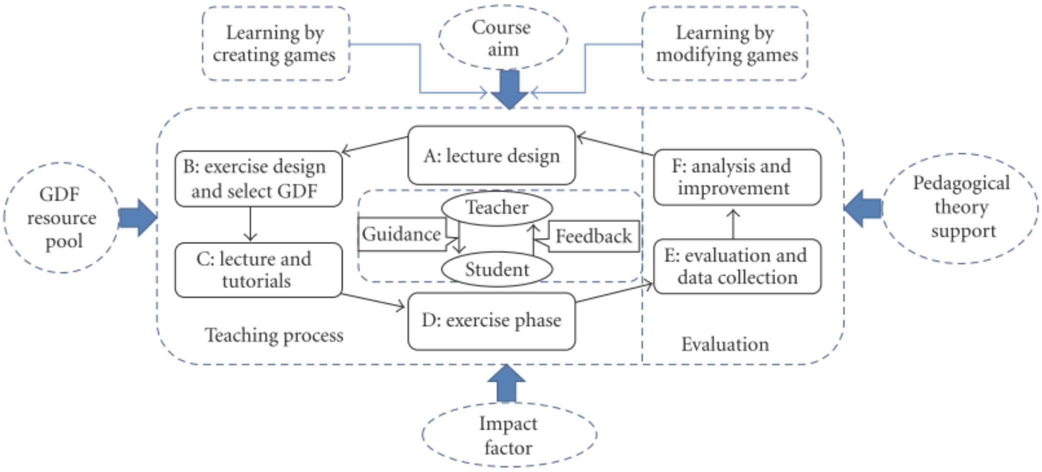


Figure 2.10.: A guideline for technical and pedagogical co-design of GDBL (Wu and Wang, 2012).

Project work in general means working in teams. Mostly, these activities follow a certain sequence (Moccozet et al., 2013). First, students decide how to approach the problem and look for a solution based on their prior knowledge. Second, the team develops roles for each member that relate well with their knowledge and complement each other in search of a common goal. Tasks and responsibilities must be shared among all members of the group to switch from individual work to teamwork. Third, they gather information from a variety of sources, analyze, and derive new knowledge from it. Collaborative work and the connection with a real world problem makes their learning valuable. Project work is always student-centered and task oriented. The final artifacts of this project work can be shared with a community, thus fostering ownership. The benefits of learning through projects include enhanced students' participation within active learning and self-learning, enhanced communication skills, addressing of a wider set of learning styles, and improved critical thinking skills. This collaboration is needed for developing the game idea, communicating, sharing, and managing assets and codes, playtesting and documentation, see previous section.

Teachers have to consider how to support collaboration and communication during the whole game production process (Ferreira et al., 2008). They must therefore stick to certain design patterns and iterative cycles (see Section 2.2.1.2) and explain game elements and rules (see Section 2.2.1.1). The project in general should foster the teamwork in producing game assets and software. Teachers have to take into account the different preferences of students, e.g., if they feel more confident in the role of developers or artists. or that students do not shy away from switching roles. Finally, the teacher has to ensure that the ideas for games are simple and clear, as well as reduce the size and complexity of the game projects.

2.2.2.3. The concept of game jams.

The idea of a game jam is to plan, design, and create games during a certain timespan while working in teams. This concept of performing game jams has been explored in three papers written by the author (Boulton et al., 2016; Petri et al., 2015; Spieler et al., 2016). The game design discipline adheres to the Constructionist approach and therefore provides a simple way of learning to code. So do game jams: they encourage working in a team, situational problem solving, and being creative.

Game jams are a way to create games under fast-paced conditions and certain constraints (Eberhardt, 2016; Deen et al., 2014). The increase in game jam events all over the world has been monumental. Their engaging and creative nature, with the aim of sharing results among players, can be seen in the high participation rate of such events: in 2013, 16,705 participants from 319 jam sites in 63 countries produced 3,248 games. By 2017, 36,401 registered participants from 701 sites in 95 countries produced 7,263 games[29] (Fowler et al., 2013). Recent studies of game jams explored their collaborative nature (Chatham et al., 2013) in combination with improvement in self-efficacy (Smith and Bowers, 2016), identified certain guidelines (Goddard et al., 2014), or investigated the motivation of jammers and their reasons for participating (Wearn and McDonald, 2016). Until now, less attention has been given to exploring game jams within an academic context e.g., students at the high-school level. Goddard, Byrne, and Mueller (Goddard et al., 2014) have identified several game jam characteristics, e.g., appropriate team size, where teams are formed (online or on-site), audience (professional or academic), timeframe (normally ranging from 24 to 48 hours (Eberhardt, 2016), occurrence (continuous or work hours), process (open, internal, or milestones), place (e.g., co-located), awards (for games or pace), constraints and submission (digital or presentations). The essential factor to frame a game jam is to define constraints for space and scope like a given theme or additional diversifiers, e.g., a local multiplayer mode or use of materials found in the public domain. These diversifiers can provide small additional sub-goals to aim for[30]. All rules push participants to be fast, think creatively, work collaboratively, and finish a game within a given deadline (Kaitila, 2012). The theme is kept secret until the beginning of the jam to ensure equal conditions and to encourage the creativity and problem solving skills by forming a game that suits the game. Preston, et al. (Preston et al., 2012) characterized a typical game jammer at the most popular game jam event: the worldwide Global Game Jam (Fowler et al., 2013), which plays a significant role in research. The participants are mostly male and already advanced in various areas (knowledge in at least one programming language or game-making software), with the motivation to meet potential business partners or to sharpen skills. Tools normally used in professional game jams (Suddaby, 2013), like the game engine Unity3d[31] or the computer graphics software blender[32] are either difficult to learn for young students or not taught in schools. This fact could lead to social pressure for novice developers or women to skip such events due to their lack of prior knowledge or access to these tools or sense of belonging (see next section). Another common issue (Jaffa, 2016) with game jams is that that jam participants need their own hardware and tools to create their projects which therefore puts participants without the ability to make their own tools at a disadvantage. Allowing game jams with visual programming tools is a promising concept that can be easily transferred to a classroom setting. In an academic setting, game jams allow students with common goals to work together while expressing individual ideas and creativity (Chatham et al., 2013). Academic game jams are a kind of project work that fosters collaboration; at the same time, these jams result in understanding learning content from different subjects (Chandrasekaran et al., 2012). Therefore, game jams cover various game-making disciplines, like programming, art, literacy, and design, and support learning by doing. The theme can center on school topics, where factors such as learning achievement, engagement, and persistence are important (Goddard et al., 2014). Game jams can contribute to improve learner's overall motivations and productivity (Domínguez et al., 2013). Therefore, learners are provided with initial goals, followed by additional sub-goals within the gaming periods. Within a school context, it is important that game

[29] Global Game Jam: https://globalgamejam.org/history

[30] Global Game Jam diversifiers: http://globalgamejam.org/global-game-jam-diversifiers

[31] Unity3D: https://unity3d.com/

[32] Blender 3D Software: https://www.blender.org/

jams focus on networking, collaboration, and engagement, and provide a positive first contact with coding tools. By staying within certain rules and constraints, the learners were encouraged to be fast, cut corners, think outside the box, and finish a game within a fixed timeframe (Deen et al., 2014; Kaitila, 2012).

To conclude, game jams are a entertaining way to create games and to motivate extrinsically. During the NOLB project GDBL and game jam approaches have been integrated in formal learning situations and their effectiveness has been evaluated. What seems to be a promising opportunity for all students raises the question of whether such game based concepts also help to fix the gender gap of women in IT related fields. Thus, the next sections provide an overview about the apparent gender-gap in technology fields and universities and broach the issue of female teenagers and coding.

2.3. Women in IC-Technology

Although promoting gender equality is a long-standing policy which all European countries place on their agendas, gender-based discrimination still poses barriers in several areas. As already explained in the previous section (see Section 2.1), the acquisition of digital skills is more important than ever and represents a key professional qualification. In this section, first models of the population pyramid are presented to show the dramatic impact of the low number of women in IT in the future. Next, the author tries to explain why ICT is an exception in underrepresentation of women by presenting a brief history about women and technology. After this, related data is presented to illustrate the current situation of women in ICT (industry and university level) worldwide, in Europe, and in Austria/Graz and further explains why it is essential to have women in this emerging sector.

There is a great potential for young women to counteract the acute shortage of qualified professionals in ICT (Tandon, 2012). However, the absence of females who are interested in ICT fields can be observed at all levels of education as well as in the industry. Taking a closer look at the different shapes of the population pyramid from selected countries and continents (Europe, Austria, Central Asia, and Africa) illustrates the problem these societies will face within the next several years (PopulationPyramid.Net, 2018) (see Figure 2.11). The population pyramid can take various forms, e.g., the form of an urn, or a drop (Boucher, 2016).

The population pyramids of Europe and Austria are very similar and both have the form of an urn. Many developed countries have this form of age structure, with a low birth rate on one the bottom and an ageing population on the top. At the same time, the younger age groups are decreasing from year to year. This phenomenon is usually called obsolescence. The shape of the population pyramid of Central Asia is similar to a drop shape, but also has a slight increase in the fertility rate. The drop shape is typical for industrialized countries, large cities, and especially for city centres. The reason is that most center districts are not very attractive to families or elderly people, but are preferred by young adults. The population of Africa shows the shape of a pyramid. Countries with this kind of population have a constantly high or even rising birth rate and an exponentially spreading base. This is typical for a low life expectancy and an early, high mortality rate across all ages. Such pyramid forms are common in developing and poor countries.

What does this mean for the future? Over the next several years, students born during the low years in Europe or developed countries of 1990-2010 will graduate from high school, thus entering their professional careers or university. Consequently, those will not replace the people retiring, meaning

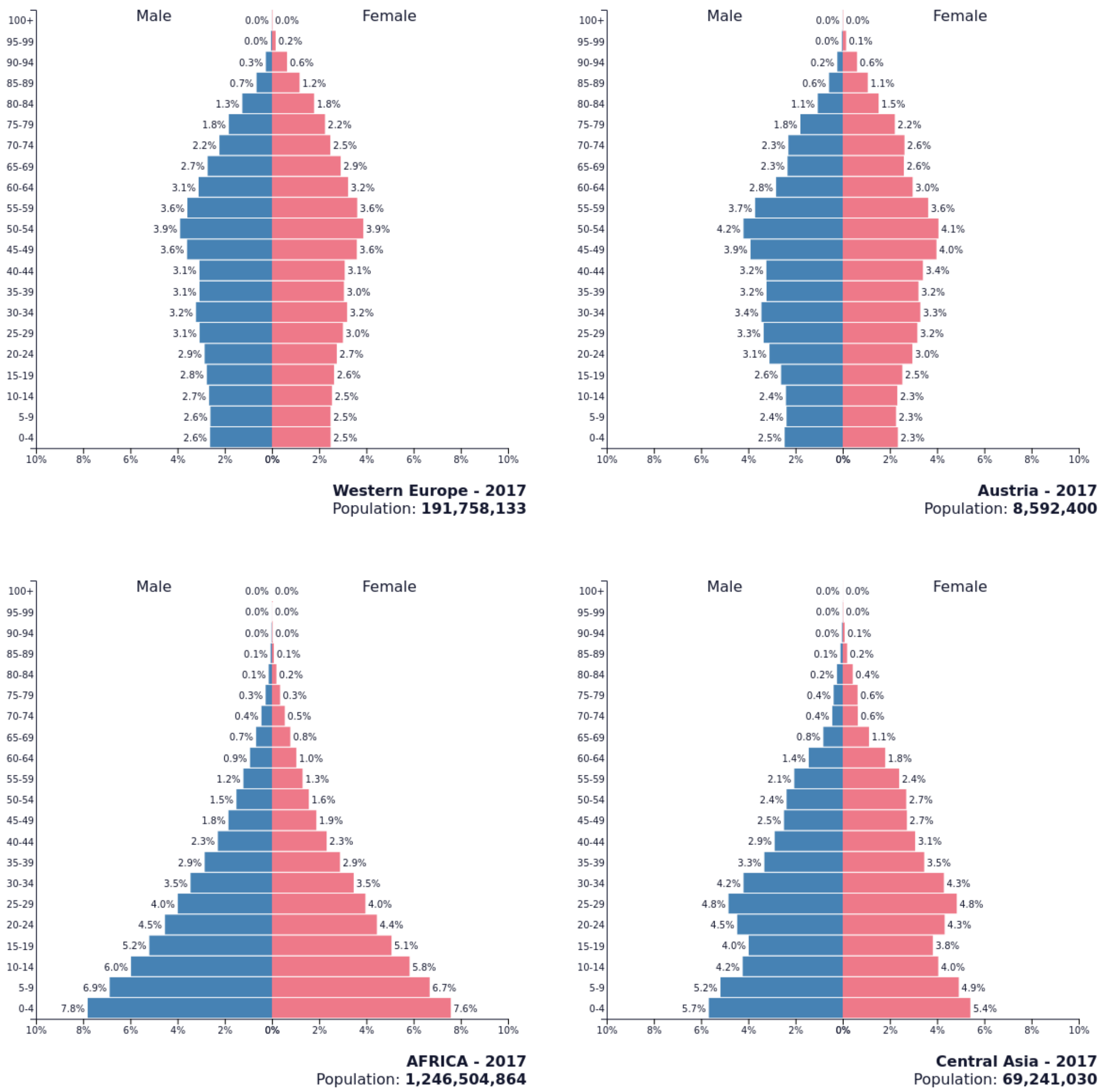


Figure 2.11.: Population pyramid of Western Europe, Austria, Central Asia, and Africa (PopulationPyramid.Net, 2018)

that the workforce will continue to decline over the coming 20 years to 77% (Schipfer, 2005). Moreover, 65% of the employees will be under 40 years of age. At the same time, the demand for IT professionals will be increasing with a dramatic impact on the industrial and economic site. While the population will increase, especially in Africa and Asia, the rest of the world will add only some millions to the total population (Grant et al., 2004). Thus, the declines in human capital will be centered in Western European countries, which potentially reduces productivity. Moreover, companies will struggle to hire young people as the population will get significantly older.

To counteract this, it is necessary to inspire more young women to aspire for careers in ICT. To encourage more women to work to increase the labor force, especially in the technology sector, is a promising approach to contributing toward the longer-term stability of the country. However, on the one hand, many young women are faced with the decision of choosing between their careers and their families. On the other hand, they may have less time to devote to having children, which then may decrease the overall fertility rate of the country again. Since this thesis is focused on how to attract female teenagers to ICT, it is not the aim to describe all possible barriers and challenges women are faced with. But of course, it is one impact that contributes to the lower number in women in the workforce and also influences the gender gap in ICT. However, even outside of the general decline of the population in industrial countries, the current number of women in ICT is already very low. Nevertheless, that was not always the case; see a brief history in Figure 2.12 of women in computing.

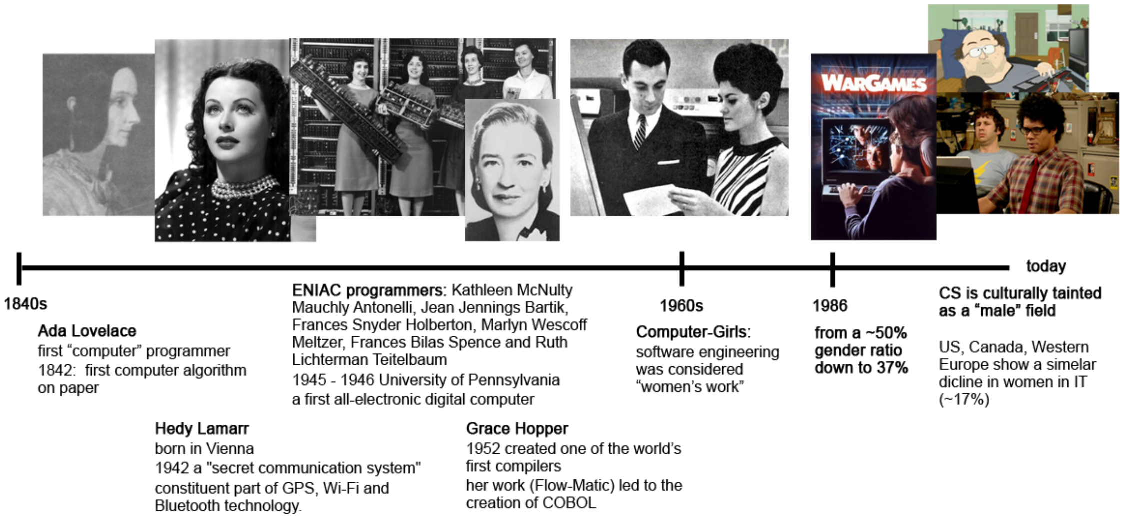


Figure 2.12.: Women were the first software engineers (The National Center for Women & Information Technology (NCWIT), 2016; Ashcraft et al., 2016). Ada Lovelace 1842 (OCLC, 2017), Hedy Lamarr 1942 (Barnett, 2011), ENIAC (McNulty et al., 1997), Computer-Girls (Harford, 2017; O'Connor, 2017)

At the beginning of the 20th century, computers and computer programmers were seen as something negative and strongly associated with women and low-paid secretary occupations. This image has changed dramatically over the last several decades. Researchers have tried to explain why there was this huge decline from approximately 50% to 37% in 1986 to a mere 17% today (The National Center for Women & Information Technology (NCWIT), 2016; Ashcraft et al., 2016). Possible reasons to describe this phenomena are largely rooted in pop culture and targeted marketing toward boys. For instance, movies in the 80s established a stereotype of the "computer programmer" (e.g., "WarGames"), the development of video game consoles and games targets mostly young boys, and IT jobs became more lucrative or cool, which appealed to boys and men hoping to score high paying jobs, prestige, and technological knowhow (also known as the "Mark Zuckerberg effect" (Kumar, 2011)). This will be discussed in more detail in Section 2.4.1 with a focus on the teenage girls of today.

In 2018, women in ICT are still facing significant barriers in the workplace. According to the statistics of the World Bank (The World Bank, 2017), women account for approximately 40% of the workforce worldwide. In Western Countries the average labor force of women is 46.55% (Austria, Spain, Germany, UK and Switzerland), but the tech field is notoriously male-dominated at all levels (Wisniewski, 2017)(see Figure 2.14 (Krook, 2018)), with the most differences in the higher levels. In the tech industry, only 21% of the executives are women.

Numbers show that during the past decade, women's positions in ICT professions have not improved and women's employment in ICT occupations increased slower than men's employment (Lamborelle and Fernandez, 2016). In 2016, only 18% of the tech workforce in the EU member states was female (see Figure 2.13).

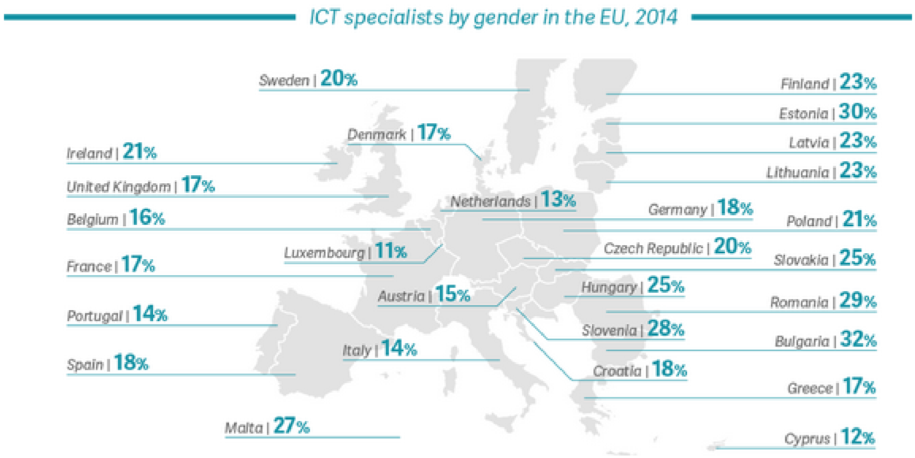


Figure 2.13.: Women in ICT: How do EU member states measure up? (Lamborelle and Fernandez, 2016)

The figure shows that the countries with the most pronounced gender inequality are the Benelux countries (Belgium, Netherlands, and Luxembourg: 13%), Italy, and Portugal, with 13% women in the tech sector (Lamborelle and Fernandez, 2016; Eurostat, 2017). Bulgaria has the highest proportion of female ICT specialists (32%), closely followed by Estonia (30%) and Romania (29%). The percentage of female ICT specialists reaches only 15% in Austria, 18% in Germany and Spain, and 17% in the UK. A more recent study from 2018 (European Statistics, 2018) summarizes all science and engineering jobs in the EU (60% were men and 40% women). But again, a similar picture: Women are underrepresented in Luxembourg (25%), Finland (28%), Hungary (31%), Austria (32%), and Germany (33%). The European Commission plans to close the large ICT gender gap, e.g., with policy and funding programmes (European Commission [2], 2016) like Grand Coalition for Digital Jobs[33], or Horizon 2020 which has the requirements to include women in evaluation panels and scientific consortia.

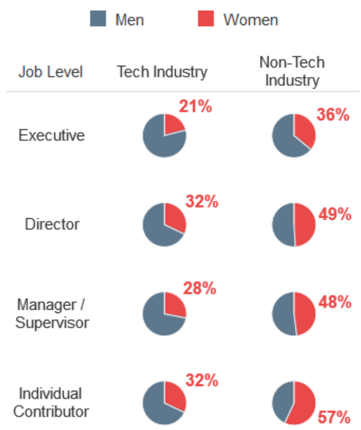


Figure 2.14.: Tech industry: a male dominated sector (Krook, 2018)

[33]Grand Coalition for Digital Jobs: https://ec.europa.eu/digital-single-market/key-priorities-grand-coalition

In other developed countries, the picture is the same. In the US in 2016, 57% of the workforce was female but only 26% of the women were working in computing occupations (NCWIT, 2017) and 56% of women who enter the field leave for other careers (IT Manager Daily, 2018). The same is true for Australia (46.2% women workforce, 28-31% of women in the roles of technology (Charleston, 2017)). Another alarming situation is the digital gap of women and ICT in Africa (Buskens and Webb, 2014), e.g., in Kafanchan or in northern Nigeria. Africa has recently seen rapid growth in Internet access but females are vastly underrepresented in this sector.

The gender gap in IT is not a worldwide phenomenon and also does not exist in such large amounts in eastern states, like Romania, Bulgaria, or Latvia (Fiscutean, 2017). Here, research shows that if the government minimizes or equalizes the choice and opportunities, i.e., all people have to work regardless of their gender, they were pushed equally into engineering and science occupations and equally paid as well; it results in more gender-neutral distribution across professions (Sanders, 2005). This is the result of the communist regime, with a huge demand in workforce, but also more women in these states occupy management positions (Fiscutean, 2017). Even working hours are more flexible; however, women are the ones who do childcare and other unpaid work (Ferrant et al., 2014). Another assumption is that the culture plays an important role (see India, Armenia (Gharibyan and Gunsaulus, 2006), or Vietnam (Shillabeer and Jackson, 2013)). For example, in the Arabic population in Israel, 61% of the students in CS classes are female. What is at first surprising or even pleasant is the result of the minimal options available to young women in those countries (Ripley, 2017). However, a closer look shows that countries that have the most female college graduates in STEM fields were also some of the least gender-equal countries (Stoet and Geary, 2018). The article states (Ripley, 2017), "A boy doesn't need to study hard to have a good job. But a girl needs to work hard to get a respectable job." In addition, the overall proportion of the female workforce relative to the overall labor force is very small across Arabic countries (The World Bank, 2017), see Saudi Arabia (16.19%), or Yemen (7.88%). Statistics from Asia show mixed numbers. On the one hand, in Malaysia (60% female students in CS programs) or Chang Gung University in Taiwan and Mahidol University in Thailand (both 48% female students in CS programs), no gender gap in IT is present (Ong, 2016). On the other hand, at the Hiroshima University in Japan, only 4% of the students are female. In India, the problem women face is a more general barrier to work than a gender gap in a specific sector (Oliver, 2017). To conclude, a comparison of "women and IT" across cultures on one dimension raises more open questions than it brings solutions. It seems that countries that empower women, empower them more to choose the career they want to or to choose other fields if they are not interested in STEM.

Many believe that there is no obvious reason why there are less women in tech industry than men and that it must be something "biological". But two arguments hold strongly against this theory: the historical and cultural differences clearly show that biology is not a factor. Some literature argues that the truth lies somewhere in the middle and that there is no single answer which explains the current situation. Despite this, a majority of people believe that in the future more women will occupy the tech industry regardless of current trends, but a steady decline of women in IT and other such factors do not support this argument (Flabbi et al., 2014).

> "I have very personally felt the overwhelming loneliness, self-doubt, and frustration that often comes with the minority status of a woman in engineering. As much as I can help others get through or avoid those difficult stretches that I myself had to weather, I'd like to. As a bonus, the more women (and minorities) that enter and don't leave the field, the better it all gets for everyone, including me!"
> Tracy Chou, Software Engineer at Pinterest

> "I think it's very important to get more women into computing. My slogan is: Computing is too important to be left to men."
> Karen Sparck Jones, Professor at Cambridge Computer Laboratory

A lot of studies refer to the imbalances or inequity in CS education system, but differences are also visible in recruiting and hiring (UK 2014: only one-in-twenty job applicants were women (Burn-Callander, 2014)), in the gender pay-gap (women earn 59% of what men earn (Wisniewski, 2017)), and retaining (US 2014: 45% are more likely than men to leave in their first year (Hewlett and Sherbin, 2014; Ashcraft et al., 2016)). To limit the scope of the thesis, these issues will not be discussed further. The focus of the next sections is to describe the problem a lack of of gender diversity in the educational pipeline.

2.3.1. The gender gap in CS-deegrees.

The underrepresentation of women in IT in the Western World is certainly the result of their overwhelming underrepresentation in CS-related degree programs. The number of female students who choose a technical degree course is far below the average number of males. This is true especially for Western Countries (Lamborelle and Fernandez, 2016; eMarketer, 2015; The Women's Engineering Society (WES), 2017; Catalyst, 2016; Google, 2016), whereas again Eastern Europe has the highest number of women in Europe enrolled in STEM degrees (but only 8.13% enroll in ICT degrees) (Unesco, 2015). To phrase it differently, out of 1,000 women with BA degrees in Europe, only 29 hold degrees in an ICT-related field.

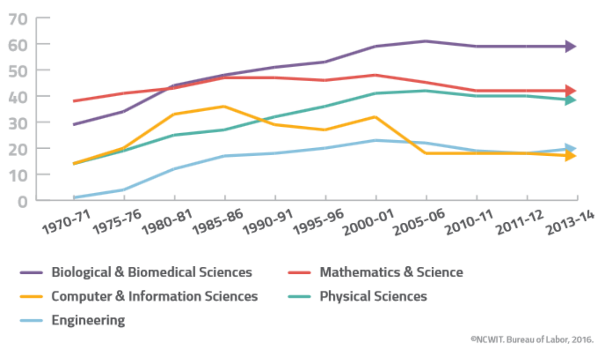


Figure 2.15.: US: Female percentage of select STEM undergraduate degree recipients (Ashcraft et al., 2016).

The term "leaky pipeline" is described in (Dee et al., 2009) and explained best by an example of the UK. All undergrads are educated in ICT related technology, as already explained in the previous section (see Section 2.1.1). This means that approximately 50% of the undergraduates who receive ICT education are female whereas only 17% of the workforce in the UK ICT field consists of women (Lamborelle and Fernandez, 2016) and only 15.8% of engineering and technology undergraduates are female in the UK (The Women's Engineering Society (WES), 2017). However, the ICT curriculum in the UK changed 4 years ago. Every year, the Federal Statistical Office in Germany publishes the newest numbers of first-year students and students in general (German Federal Statistical Office, 2017). In 2017, the number of students in total was higher than ever (+ 1.5%). However, in comparison, the number of first-year students in computer science shows a significantly decrease (-4.1%) and the number of first-year female students has even declined at 8.8%.

Statistics from the US (Espinosa, 2015) in 2014 showed that women earn only 19% of engineering and 18% of computer science bachelor's degrees. Figure 2.15 shows women's participation in science increased significantly but decreased in CS (Ashcraft et al., 2016). In Canada, a study about career interest (Shillabeer and Jackson, 2013) showed that the lack of interest of women reduced the percentage of people indicating that they were interested or very interested in "High Tech/Computers" to 20%. In Australia (Catalyst, 2016) in 2015, only 15.5% of those enrolled in information technology, engineering, and related technologies were women. As described in the previous section, countries with higher gender inequality mostly have a higher number of women in STEM fields due to reasons which are not always fair or under the best conditions (Stoet and Geary, 2018). Figure 2.16 holds a scatterplot of countries that shows the number of female STEM graduates and their Global Gender Gap Index (y-axis) (Khazan, 2018).

Research on women at universities shows that some of the female students believe that their gender is incompatible with CS (e.g., it makes them question their ability or position), while others feel that being female is a motivator to challenge the stereotypes (Denner et al., 2015), and some mention in tech they have the feeling of being financially secure and on equal footing with men (Stoet and Geary, 2018). Since the focus of this thesis is to reinforce females in their early years, more research towards female students and their interests, confidence, and motivation in middle schools is presented subsequently in Section 2.4. But first, the author tries to answer why it is important to have more women in tech fields in general.

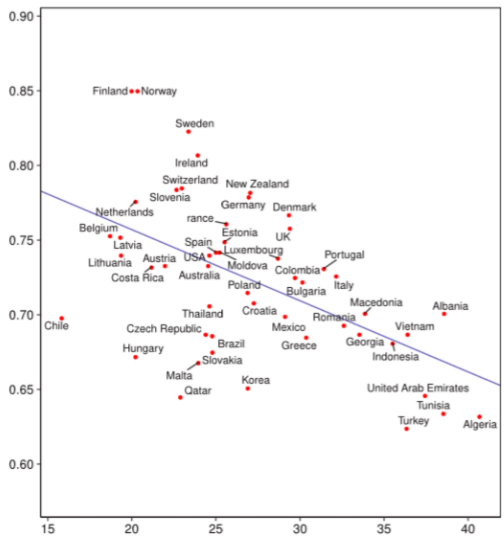


Figure 2.16.: Scatterplot of countries that shows their number of female STEM graduates and the Global Gender Gap Index (y-axis) (Khazan, 2018).

2.3.2. Situation at universities in Austria and Graz.

In Austria, the total percentage of women at public universities in STEM fields is 34% (Summer Term 2016) and in all other fields at university, the overall percentage of women counts for 61% (Binder et al., 2017). At universities of applied science ("Fachhochschulen"), the proportion of women in STEM subjects is even lower (23%). Furthermore, at public universities, about 36% of the degrees in STEM studies are earned by women (FH: 23%), in other fields the percentage reaches 65% (FH: 64%; see Table 2.1).

Table 2.1.: Gender distribution at all universities in Austria. Studies in the Winter Term 2015/16. University degrees in the year 2014/15.Hochschulstatistik (BMWFW) (Binder et al., 2017)

		studies		**university degree**	
		women	**man**	**woman**	**man**
FH	STEM total	23%	77%	23%	77%
	other fields	64%	36%	64%	36%
Uni	STEM total	34%	66%	36%	64%
	other fields	61%	39%	65%	35%

A lower number of women can be recognized especially in the degree programs of computer science (university: 16%, FH: 15%) as well as engineering and engineering programs (university: 18%, FH: 21%; see Figure 2.17). The proportion of women in architecture and civil engineering (42%), in chemistry (47%), and especially in biosciences (65%) is in comparison relatively high (see Figure 2.17).

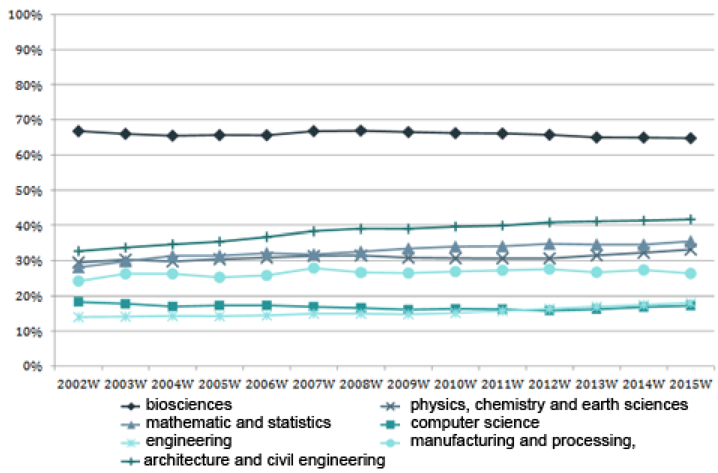


Figure 2.17.: Development of the proportion of women in STEM studies at public universities by fields. Bachelor's, master's, and diploma programs are shown in Winter Terms. University records (BMWFW) (Binder et al., 2017)

In total, all male and female students in STEM bachelor programs have the same success rates of about one third (Binder et al., 2017). However, more men (16%) further enroll for a graduate study (master's or doctoral program) than women (10%). Degree programs of other fields are more likely to be completed by women. Here, it is apparent that the higher the proportion of women in a subject, the more frequently women complete their studies compared to men, and likewise (Dornmayr and Winkler, 2016). The only exception in success rates and gender is the study of computer science: Here, women have a 10%-points lower success rate than men. One explanation of this gender related success rate is the differences in the school education. Overall, women in STEM studies more often have an AHS-Matura (66% vs. m 50%) and more rarely an HTL-Matura than men (8% vs. men's 37%). To conclude, the success rate of female students in the STEM area with an HTL-Matura degree is much higher than those of students with an AHS-Matura degree.

The percentage of female students at Graz University of Technology (TU Graz) in Austria for the winter term 2017/2018 reaches just 24.44%[34]. This number increases slightly every year (Summer Term 2017: 23.70%) but it is mainly due to the high amount of female first semester students in architecture, biomedical engineering, and chemistry. A closer look at the percentage of women's bachelor's degrees in computer science shows a percentage of 12.95%. The gender distribution at the TU Graz in the different levels continues to show inequality (Koffler, 2018). Among students, women are in the minority (78% men) and among the graduates almost 80% are male. The proportion of women falls to 25% for doctoral graduates. The amount of female professors clearly reflects this unbalanced gender ratio. At the upper levels of career opportunities within universities, the proportion of women falls to 26% assistant professors and for professors even lower to 7% with means that from 118 professors at TU Graz only 8 are women (see Figure 2.18).

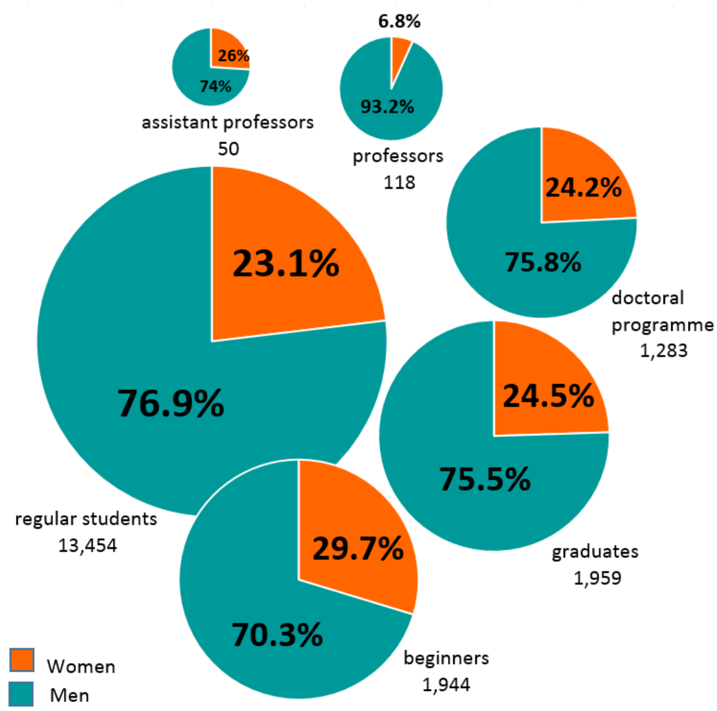


Figure 2.18.: TU Graz: students and staff 2017 (Koffler, 2018).

An examination of the number of students regarding the degree programs "Software Development" and "Business Management" at TU Graz is presented in Table 2.2.

Table 2.2.: Percentage of female students in Software Development and Business Management at Graz University of Technology.

	% of females 1st semester	% of females 4th semester	% of retention
2011	10%	6%	54%
2012	13%	7%	51%
2013	25%	18%	73%
2014	23%	18%	77%

[34]TU Graz student statistics: https://online.tugraz.at/tug$_online/Studierendenstatistik.html$

This table points out two major problems we face at TU Graz. First, a very slight increase in the number of female students over the years and second, an increase in the retention rate, which further represents a decrease in the number of finished degrees.

For the representation of the phenomenon of the "Leaky Pipeline", often so-called scissors diagrams are used (Eckstein, 2014). Figure 2.19 shows the career path at University of Graz. Here the number of female university assistants is relatively the same as the number of men, but when it comes to the number of female doctoral candidates and especially the number of female professors, the gender gap is clearly visible.

Figure 2.19.: "Leaky Pipeline" Karl-Franzens University Graz (Eckstein, 2014).

To conclude, the situation is not surprising as it is similar to those in other developing countries. Whereas in general there are more female students at university level, the percentage of women who study computer science-related subjects is very low. Reasons why female teenagers decide not to choose CS as a major are described in Section 2.4.

2.3.3. The importance of women in ICT.

The reason why the world needs more women in ICT disciplines is not only a matter of fairness (Milgram, 2011). It is necessary because women represent over half of the world's population but are mostly underrepresented in those fields (see previous section). Girls and women have caught up with boys and men as users of technology, but not as producers of it. Our future will largely be digital and its innovation is by no means complete. Not involving women creates a situation in which improvements, projects, and research are largely done without their participation. Moreover, this means that they are not part of important decisions being made in the world of tech today and that funding will not be awarded to develop their ideas and concepts.

A number of equality policies try to ensure gender equality at all levels. According to Gildemeister (Gildemeister, 2010) "gender" is created and reproduced as a socially distinction. Thus, gender is the construction that society uses to determine which roles, behavioral descriptions, professions, and social opportunities are connected with respective gender affiliations (Bauer, 1995). Gender Mainstreaming (Europe of Council, 1998; Ketelaars, 2017) compromises the organization, development, and evaluation of the decision-making processes, with the aim that the actors involved in political shaping take the viewpoint of equality in all areas and at all levels. The contract of Amsterdam 1997 states the promoting of "equality between men and women" is an important task of European society (Office for Official Publications of the European Communities, 1997). The National Science Foundation (NSF) includes the importance of "preparing a diverse, globally engaged science, technology, engineering, and mathematics (STEM) workforce." (Iacono, 2013). In addition, similar agreements are stated in the Federal Constitutional Act of Austria 2009 (Federal Chancellery Austria, 2009) and in the Equal Treatment Act of 2004 as a cross-cutting issue (Bundesgesetzblatt I No. 66/2004, 2004). The goal is to restructure traditional thoughts and change the way culture and society is currently divided and organized. specific measures in favor of women are, for example, quantitative measures (quota: binding and voluntary), and qualitative measures (mentoring, initiatives, networks) with the goal of integrating women in existing societies. Quotes, policies, and measures are necessary, because there already exists huge hierarchical gender segregation (in leadership positions) (European Commission [3], 2013), a horizontal gender segregation (unequal distribution of gender across different work areas) and a mismatch between work performance and payment (Ferrant et al., 2014). When women take the role of technology leaders, it can break down personal identity barriers such as gender role stereotyping and discrimination.

Gender equality is positive: A number of studies show that groups of employees with diverse backgrounds, interests, or cultures leads to better results and that they come up with more "neutral" or "critical" perspectives (Tsan et al., 2016; Cukier et al., 2002; Sanders, 2005; eMarketer, 2015). A study that observed more than 100 team of 21 companies concluded that teams with equal numbers of men and women were more likely to experiment, be creative, and share knowledge (NCWIT, 2015); additionally, women in tech teams boost problem-solving and creativity in teams (IT Manager Daily, 2018). Moreover, one study has found that adult groups which consist of women outperform groups which consist of male members in a collaborative task (Woolley et al., 2010). Increasing the number of women in organizations leads to an increased sense of loyalty and boosts staff motivation and satisfaction, which also improves the image of a company (Olgiati and Shapiro, 2002). McKinsey & Company Inc. (Hunt et al., 2015) suggests that successful companies make gender diversity a priority. Companies that have made significant efforts to diversity are the most successful and an analysis of 20,000 companies showed that successful tech startups have twice as many women in senior positions (Hunt et al., 2015). Equal opportunities are accompanied with higher economic performance, which leads to improved economic efficiency (Flabbi et al., 2014; Julizaerma and Sori, 2012), better economic numbers in general (Hunt et al., 2015; The National Center for Women & Information Technology (NCWIT), 2016), and increases the value of the company overall (Campbell and Vera, 2010; Carter et al., 2003). Gender equality and equal treatment increase the sustainability of organizational decisions (Eagly et al., 2014) and it has been shown that women make less unethical decisions at work overall (Kish-Gephart et al., 2010). The European Commission's report that "The Business Case of Diversity" showed benefits that companies gained ranged from reduced absenteeism, reduced employee turnover, and a stronger corporate image for companies that include diversity practices in their business. Frieze and Quesenberry (Frieze and Quesenberry, 2015) add that just acquiring more women does not improve the overall situation (e.g., only to meet the quota which may leads to a high retiring rate again) but inclusive environments are the key.

All people can benefit from environments, policies, and cultural beliefs that reinforce inclusivity and equity. Staut and Camp (Stout and Camp, 2014) mentioned two more viewpoints. First, from the humanitarian standpoint, equality means that all humans should have the same opportunities, and second, from a capitalistic perspective, we must consistently ensure a homogeneous set of experiences among the workforce. To conclude, everybody has to understand that this is not a “women’s issue”, it is a “people’s issue”, and it should be in the interest of the whole computing world, rather than in the interest of any specific underrepresented group to enhance diversity in general (Ketelaars, 2017).

2.4. Closing the Gender Gap in ICT in Female Teenagers

Paying closer attention towards diversity in ICT raises the question of why there are not more women in this industry, in STEM/ICT degree programs at universities overall, or even in voluntarily offered computer courses at high schools or off-school initiatives. This topic has been already discussed for decades, with researchers wondering what prevents young women from choosing an ICT career. A lot of recent literature addresses this bias or tries to explain diverse reasons why the gender gap in tech exists. On the one hand, it has been investigated if teenage girls have different (pre-) perceptions, strategies, and qualities (Dasgupta and Stout, 2014; Castillo et al., 2014; Cukier et al., 2002; Unfried et al., 2015; Khan and Luxton-Reilly, 2016; Eccles, 2007), or a lower self-efficacy or sense of belonging or more negative attributes in regard to computers than boys (Beyer et al., 2002; Pajares, 2005; Beam et al., 2002; Veilleux et al., 2013; Eagly and Wood, 2012; Heilman, 1983; Galdi et al., 2014; Young et al., 2013; Good et al., 2012). On the other hand, the literature assumes 1) that girls are being less interested, motivated, or engaged in STEM/ICT disciplines (Gabay-Egozi et al., 2015; Medel and Pournaghshband, 2017; Hulsey et al., 2014; Milgram, 2011; Tsan et al., 2016; Davies et al., 2014; Zagami et al., 2015; Carter, 2006; Appianing and Eck, 2015; Baumeister and Leary, 1995; Walton and Cohen, 2007; Eccles, 2011), and, 2) that missing female role models or mentors (Young et al., 2013; Galdi et al., 2014; Young et al., 2013; Mason et al., 1991; WiCS Avocacy Council, 2015; Ong, 2016; Oliver, 2017; Next Generation Recruitment, 2018; Back et al., 2011; Cohoon, 2011; Lockwood, 2006; Ko and Davis, 2017; Clarke-Midura et al., 2016; Zaidi et al., 2016; Wood and Mayo-Wilson, 2012; Holliday and Luginbuhl, 2004; Townsend, 2002; The Association for Women in Science/AWIS, 2017), or the existing stereotypes about geek culture influence their career choices (Cheryan et al., 2013; Bartilla and Köppe, 2016; Stout et al., 2011; Cheryan et al., 2011; UNESCO Asia-Pacific, 2017; Gabay-Egozi et al., 2015; Monitise Group Limited, 2014; Vervecken and Hannover, 2012; Matlin, 1999; Master et al., 2016; Cheryan et al., 2015, 2009). Researchers are certain that girls’ interest in IT changes early in their adolescence, either in middle school or the early years of high school between the ages of 12 and 15 years old (Sadler et al., 2012; Kafai, 2006; RTE, 2016; Shillabeer and Jackson, 2013; Cadinu et al., 2005; Cukier et al., 2002; Gabay-Egozi et al., 2015; Khan and Luxton-Reilly, 2016; Zagami et al., 2015; Beyer et al., 2003; Unfried et al., 2015; Mann and Diprete, 2013; Ochsner, 2015; Monitise Group Limited, 2014; Master et al., 2016; Ko and Davis, 2017). These are exactly the years where first career choices and low levels of participation in STEM subjects occur.

To explain some reasons for the gender gap in ICT, this section first explains first critical factors from the literature which may exclude females from traditional computer science curricula. Second, promising ways to close this gap by setting the focus on intrinsic/extrinsic motivators for teenage girls are presented. Finally, solutions and existing initiatives to attract female teenagers are analyzed as well as girls gaming behaviors with the goal of shaping inclusive and gender sensitive learning situations. In addition, three expert interviews have been conducted; insights are included in this section as well.

The expert interviews have been conducted with:

- Dr. Marita Haas

PostDoc at the Institute of Management Science, Vienna University of Technology

- Prof. Dr. Libora Oates-Indruchová

Professor of Gender Sociology, Institute of Sociology, University of Graz

- Dipl.-Ing. Gudrun Haage and Florian Ungerböck, BSc MSc

Gender Equality and Affirmative Action Office, University of Technology Graz Lead of CoMaed girls-only initiative

2.4.1. The gender gap: a social construct.

A majority of people believe that in the future more girls will enroll in STEM related subjects anyway (Flabbi et al., 2014). However, current numbers are alarming, as stated in the previous section. High school is a critical place where students decide their future careers, develop a more realistic picture of their future jobs, and assess their career-relevant abilities (Charles and Bradley, 2009; Vervecken et al., 2013; Vervecken and Hannover, 2015; Chen, 2007; Carter, 2006; Appiaining and Eck, 2015). The assumptions about why girls are less interested in ICT have not significantly changed in recent years. The reasons highlighted in 2005 (Cadinu et al., 2005) or 2002 (Cukier et al., 2002) seem not to be outdated and can be found in most of the literature today (Cheryan et al., 2013; Zagami et al., 2015; Alvarado et al., 2017; Wong and Kemp, 2017; Ko and Davis, 2017; Zaidi et al., 2016; Ramos and Rojas-Rajs, 2016). For example, this research review (Clayton et al., 2009) describes that boys are more likely to spend time with solving difficult computer problems and seem to embrace ICT more. In contrast, girls take more personal responsibility for their failures, see them as a lack of ability, and for them technology is more a means to an end. To conclude, the literature is full of findings that document women's low experience levels towards computer sciences, their negative attitudes, and their fear to fail in computer science subjects compared to their male colleagues. Thus, the research focuses on female deficits in ICT. However, what if these are not deficits but just *different approaches* to reach the goal, and *different points of view*? This means that if classes allow a *diverse range of outcomes* and include the possibility of *different understandings of the discipline*, girls can reinforced so that they become more interested in coding.

Therefore, this section explains first several critical factors from the literature which may explain the mostly socially constructed gender gap in IT. These social constructs exclude females from traditional, mostly male-oriented computer science curricula. As a result, an image of the stereotyped "Girl in Tech" is presented and which shows how females are intrinsically not reinforced for coding activities or to pursue and persist in IT. This method is called "dramatization" of gender (Engler and Wieland, 1995), with a focus on separate groups by gender in order to build sensitivity and awareness. The risk of this method is a hasty assumption of dramatic differences between males and females, which would strengthen stereotypes and deepen the focus on a bi-gender system as opposed to a diverse one. The theory of "strategic essentialism" (Spivak, 1990) states that such categories are important and also, even if they are socially constructed, they are not less effective. Thus, it is proposed that taking categories seriously must happen in order to analyze social inequality and to reinforce minorities as a result (Scambor and Scambor, 2012).

2.4.1.1. Stereotypes, preconceptions, and the (male) dominated tech-sector.

Stereotypes and preconceptions influence all people and produce (mis-)representation (Matlin, 1999). They describe specific behaviors, attitudes, and capabilities (Eagly and Wood, 2012), which are associated with a certain profession, race or gender. Studies show that women at all levels (teenage girls, undergraduate female STEM students, and even successful female scientists) are weakened by stereotypes in their field (Galdi et al., 2014; Young et al., 2013; Mason et al., 1991). Stereotypes can be transmitted through media, role models, or academic environments (Cheryan et al., 2015). While making educational choices, male and female students may follow these perceptions of an appropriate choice for their gender rather than decide in regard to their interests (Gabay-Egozi et al., 2015).

Female students must feel a sense of belonging to people in the field and it is important how well the career plans align with their personal goals and values (Stout and Camp, 2014). They are more likely to value fields stereotyped as appropriate for their gender. It can be already seen in young girls that they have stereotypes of computer scientists in their mind and see them, e.g., as socially isolated young men whose genius is the result of genetics rather than knowledge learned at school. Female students who actually take computer science classes do not have the same prejudices as young females who get their ideas from pop culture. Thus, teenage girls struggle more to decide whether they fit into ICT careers than men do, while not considering that computer science students can of course have broad interests outside of ICT as well. Disinterest can lead to unrealistic perceptions about that field (Khan and Luxton-Reilly, 2016; Carter, 2006). Thus, it also matters what students think about the discipline itself, e.g., its structure, usefulness, and how it can be learned (Dee et al., 2009). Computer science, for instance, is not only programming like already discussed in previous section (see Section 2.1.4), but more likely a tool for solving problems or creating new ideas. Some teenagers never quite understand what computer science is and how it relates to algorithmic thinking or problem-solving (Giannakos et al., 2014). The question they may ask themselves is "How likely is it that I need STEM knowledge in my future career?". This decision influences his or her motivation and persistence within an academic track. Female students seem to have more negative attitudes towards STEM classes, thus there is a much higher possibility to negatively rate this expectancy value. A research study in public highs schools located in Los Angeles investigated students' decisions (over 200 interviews) to study computer science as a major and evaluated both the structural and the psychological factors (Goode et al., 2006). Results showed that factors which discourage females from computer science courses included a) the image of the computer science subject itself, e.g., the lone programmer, b) that their familial or social networks are not technology-oriented and they do not understand the field at all, and c) they do not see the connection of how computer science could support their academic and career plans. The last point is very critical — most found programming interesting but the decision not to pursue computer science was more a strategic decision for them, e.g., to fulfill the high school graduation requirements rather than play around with computers.

Research has shown that the stereotype threat can harm the performance and sense of belonging of all individuals if they are confronted with a stereotype-based expectation of poor performance. In one study (Cadinu et al., 2005) students were confronted with the conclusion of a research study before writing a mathematical test: "Women do worse in mathematics in comparison to men". As a result, in the female students of the test group, more negative thoughts regarding their inability to perform the test could be noticed. Spencer, Steele, and Quinn (Spencer et al., 1999) goes one-step further and describes that if students have to verify their gender at the beginning of a test, it is already enough of a problem to affect their grades, e.g., for men to write inferior language tests and for women to write inferior mathematics tests.

> "Studies show that if a man fails a math test at university, he fails as a person, but if a girl fails, she fails as a representative of the whole group of women and girls. [...] this makes it so difficult." Marita Haas

An important factor to mention is parental preconceptions and influences (Zagami et al., 2015; Gabay-Egozi et al., 2015) young girls are confronted with because their families exclude computing or engineering as a possible career path for their daughters. Teenage girls are less likely to choose an IT related study unless they were encouraged by their parents, by teachers, or by their peers. Support from home is essential to have positive feelings first, e.g., through tinkering, reading books about this topic, learning through friends or relatives, or first attempts to create their own programs (Goode et al., 2006). The aforementioned study (Goode et al., 2006) showed that female students are more likely to choose to enroll in programming courses when they already know somebody who is interested or has worked in the computer field. Girl's lower sense of belonging corresponds with the feeling of "Lack of Fit" or "the Sense of Not Fitting In" with computer science stereotypes (Master et al., 2016; Alvarado et al., 2017). This occurs if CS students feel that they do not receive help, or have no ability in CS or feel intimidated. If the profession does not fit to the "traditional gender model", one is not as likely to pursue or feels discriminated against by someone who does. To be socially connected and respected is a strong initial motivator, as quoted by the researchers (Baumeister and Leary, 1995; Walton and Cohen, 2007): it can "create a sense of belonging that can reinforce students' self-efficacy and connections to community that support student perceptions of their ability within the field" (Veilleux et al., 2013, p. 64). This is important in students' decision to pursue IT and helps to identify with the field (Beyer et al., 2002). Since the stereotype in IT is more associated with a male role, female teenagers are less likely to feel a sense of belonging with these stereotypes (Master et al., 2016; Cheryan, 2012).

As a female technician, it is part of one's daily life to experience gender-biased situations, e.g., during recruitment and evaluation processes, or restrictive regulations and norms. Side effects could include exclusions from networks, and may create work-family conflicts. All of these factors make girls question whether their abilities and interests are harmonious with their selected field (College and of University Women, 1992). Their sense of belonging and collaboration stands in contrast to the image of the stereotypical lonely programmer. They feel a pressure to perform well even when they try to ignore the obstacles that hinder them, whether these are real or perceived. This stress arises from the conflict between wanting to be liked while at the same time having to demonstrate competence in competition with men (Gabay-Egozi et al., 2015). Some jobs follow the "hero mentality", where men work very hard in order to fix problems whereas women are often better at finding and correcting problems before they happen (thus doing the less hard and crucial work) (Gibbs, 2014). These typical computer science labs (Master et al., 2016) can easily be adopted in their environment or in making small changes in the language people use in such environments. It would create a huge impact to make labs more gender-neutral (Stout and Camp, 2014) and to make women feel included in their work environment. According to a study, college women who were confronted with CS classrooms full of stereotypical objects (e.g., science fiction posters, stray electronic parts) say that they are less interested in CS than women who attend classes with non-stereotypically associated objects (e.g., art posters, general interest books) (Cheryan et al., 2009). This has no impact on boys. A stereotypical classroom increased girls' concerns about negative stereotypes which decreases their sense of belonging and interest in CS courses. This corresponds with other studies, e.g., that the neutrality of technological environments hides male-dominated cultures (Ramos and Rojas-Rajs, 2016).

To summarize, a sense of belonging, the perceived value of this field, and peers in STEM/ICT are a particularly strong predictor of girl's interest and motivation, whereas stereotypes and stereotyped classrooms in CS will discourage young women.

2.4.1.2. Gender inequality in education.

Education has a significant impact on gender. It starts with gender-sensitive frameworks, with teachers who are aware of gender sensitive education, or those who have had to undergo gender responsive teaching training to ensure that the learning materials are free of gender stereotypes as well (UNESCO Asia-Pacific, 2017).

For example, teaching materials, teaching designs, and interactions often (re-)produce asymmetrical gender relations and role stereotypes. In addition, a research study on more than one hundred Australian and international research publications (Zagami et al., 2015) points out that existing computer science education materials may not be as supportive for female students. Furthermore, they do not support their interests or address their preferred activities. According to this study, representation, imagery, and language that may promote gender inequality must be considered when providing teaching material for learners (Medel and Pournaghshband, 2017). It is important to place an emphasis on promoting gender equity and diversity in computing (Stout and Camp, 2014). Another assumption is that teachers also treat students differently depending on their gender (Stoet and Geary, 2018).

> "I've talked a lot with teachers that have already formed a picture at the beginning of an CS class. A picture of the boys who can already program and this picture of the few girls who decide to try it out. One approach would be to discuss this first and talk about this with the young [. . .], e.g., what do you think why are there so few girls in this course [. . .] teachers can begin to ask students: Which of you likes math? [. . .] Who is on Facebook? [. . .] I think it's just fine to think about questions that open the whole conversation up."
> Marita Haas

> "You should pretend that there is no stereotype as if it were something normal."
> Florian Ungerböck

Here, the experts are not unanimous and it is in question if girls are resigned to the stereotype threat or strengthened by making it a topic that they can confront and overcome. Here, it depends on the situation and the teacher must decide.

However, "gender competence", "gender-specific interest guidance", or "gender-sensitive and aware education" are new territories for many teachers and schools (Giltemeister and Robert, 2008). Especially in education, gender is constantly negotiated, produced, and reproduced by students as well as by teachers. Gender-sensitive teachers recognize gender-stereotyped influences on students and counteract against them. They analyze their own role and their teaching with the aim of creating equal opportunities for girls and boys (National Science Foundation, 2018; Bosch et al., 2014). To use more gender fair language it is important to make women visible and hearable in the language as well. Many examples exist of how gender mainstreaming tries to break down the walls that hinder young women to succeed in male-dominated sectors (Walby, 2005). A gender-sensitive- language particularly important in languages which have gender specific nouns or those which prefer to use the form of the masculine plural as an inclusive way, like German[35]. For these kind of languages it

[35] differences between: genderless languages, e.g., Finnish, Chinese, Latvian, natural gender languages, e.g., English, Dutch, Swedish, and grammatical gender languages, e.g., German, French, Italian

is stated that women are included in the male generic form as well or the female form is banished into a footnote (Braun, 2008; Szekeres, 2005). A gender-inclusive language is essential (Dasgupta and Stout, 2012; Sczesny et al., 2016; Horvath et al., 2016; Formanowicz et al., 2015) because 1) "Language creates pictures" and often creates a generic masculine picture (male bias). For example, this quote "A scientist in his laboratory is not a mere technician: He is also a child confronting natural phenomena that impress him as though they were fairy tales" — produces a picture of a male scientist in his laboratory, but it is actually a citation from Marie Curie, 1937. 2) For children: If typically male occupations mention both genders instead of the masculine generic, then children assess women and men equally in this occupations (with male mentions, men are rated to be more successful) and girls tend to be more interested in these professions as well (Vervecken et al., 2013; Vervecken and Hannover, 2015). 3) Job advertisements (Hentsche and Horvath, 2015; Horvath, 2015): Women and men are more likely to apply for gender-appropriate positions and jobs and organizations which are assessed more attractive to their gender. If companies/universities want to appeal to female talents and they are using masculine language (e.g., dt. "Für den Unternehmer von morgen!"[36]) or images they may present the program as generally less attractive for women. A study with 156 female students showed that with the previously mentioned advertisement, they felt no sense of belonging and had a lower intention to apply than men. 4) Image of organizations: Women feel more supported and it opens possibilities for teamwork and family/life balance. To conclude, those who write in a gender-equitable way do not exclude anyone and thus make an important contribution to actual gender equality (Vervecken and Hannover, 2012).

> "I believe sensitization is the first step, to become more aware of appreciations. For example, at school. How do I formulate my tasks? How much space do I give for discussion, who do I tell to perform certain tasks and is there any discrimination or do I give privilege to some people. It is important to question our own stereotypes." Marita Haas

In addition, Craig et al. (Craig et al., 2013) observed that the participation of girls in computer classes is not the same as those of boys: girls tend to spend more time on visual customization while boys spent more time on solving logical puzzles, and the authors point out that it is essential to consider gender differences in logical and computational skills. According to this study, it may thus be more effective to get girls interested in technology by asking them to design games rather than to focus on the learning of specific programming skills. In programming, different styles exist among genders (see next Section) but differences also exist within same-gender groups. To engage girls in coding, social relevance, storytelling, and design tasks should be part of the coding activities (McLean and Harlow, 2017). Such activities provide the opportunities to participate in problem-solving by providing learners with resources and a given problem. Teachers that only focus on computer instruction may discourage many students (of all genders) from actively participating.

Other researchers argue that girls have a low confidence in CS environments (Jonsson, 1999; Frieze and Quesenberry, 2015; Margolis et al., 2014). For example, girls tend to ask fewer questions in heterogender classes compared to classes with a larger percentage of girls. As a result, they may be left with doubts and are less confident with the material. Bad experiences lead of course to bad expectations: If young women are used to failing or even think they will fail in a specific course, they will instead choose subjects that they know they are comparatively good in (Jonsson, 1999). Other studies show that girls tend to be less self-confident regarding their beliefs about their abilities in STEM disciplines. Peer acceptance is a central point in adolescence (Dasgupta and Stout, 2014) and collaboration is particularly important; adolescence is when students exchange ideas, defend their

[36] In English: "For the entrepreneur of tomorrow"

positions, discover new ways of thinking, and experience self-confidence, which can lead to successful task completion (Dasgupta and Stout, 2014; Patrick et al., 2007). It is well known that students find STEM courses to be more meaningful when they can connect with their peers. This study (Frieze and Quesenberry, 2015) also concludes that male students have a consistently higher confidence level concerning their coding skills than their female colleagues but they explain this phenomenon with the general tendencies that women have to downplay their abilities (Margolis et al., 2014). For instance, this attitude is displayed in the statement *"I feel like everyone I know performs better than I do."* More than 50% of the women agreed whereas only 30% of the men did so (Frieze and Quesenberry, 2015). However, they found evidence of improvements of the performance during the degree program. Other studies report girls' lower self-efficacy in comparison to boys in STEM as well (Pajares, 2005). For example, a study done by Harvard (WiCS Avocacy Council, 2015) express that male programmers rated their confidence in programming with 3.3/5 after 0-6 months of experience; this number was much lower in women (2.6/5). Only after 8 years of programming, women stated the same confidence level as men after 0-1 years of programming experience. Thus, it is important to praise the work of female students and to provide recognition of their work done. Studies indicate that male students are praised more often and also ought to be rebuked as well (Shalaway, 1998). On the one hand, it is important to sufficiently honor students' achievements, and most importantly, to shape the feedback in a such way that it does not include harmful attributions or tension which can cause the fear to fail (Schwartz, 2013). On the other hand, it is not helpful to praise girls for achievements, which are left unaddressed in boys. Thus, if one is praised for her normal performance, it can damage girls' self-confidence. While boys are often praised for their talents, girls are more often praised for being hard-working rather than simply being good. The literature suggests praising the process they engage in, not how smart they are in certain tasks.

> "Direct encouragement is often missing in schools. Most girls who take computer science courses have already experienced some form of encouragement." Marita Haas

Observations and field notes within CS classrooms showed more issues of confidence (Denner et al., 2015). Female students refused to ask for help because they had the feeling that the boys in the classes were much more advanced in CS and felt inferior to their male peers. Consequently, the authors of this paper describe the classroom climate as the following:

- Girls worked together in groups, while boys worked independently around them.
- Girls asked the teacher a lot for assistance, while boys asked their male neighbour first. Thus, boys tended to copy the code from each other.

As a result, teachers who are not aware of inclusive and gender-sensitive education can create a classroom in with male students occupy the more knowledgeable position and females are always situated in the less knowledgeable position.

2.4.1.3. Absence of female mentors and role models.

Providing a mentorship and appropriate role models are two key elements of successful professional development (Stout and Camp, 2014). Mentors provide some kind of guidance (e.g., through the first semester years) and role models can inspire deeply when acquiring new knowledge. An absence of female role models is one the top five of reasons why women are underrepresented in tech (42% mention this in the findings of an ISAC study) (Wisniewski, 2017) and it is a topic cited by many other studies, for instance (Milgram, 2011). It is important that girls see female role models who have succeeded and who promote positive beliefs regarding women's abilities, which demonstrates that this job field can suit them as well. Since the tech sector is very male-dominated, most role models are also male. In the male-dominated tech field, female students need encouragement, but literature still argues if mentors must be women or if it is only important to have somebody who believes in them and encourages them (Lockwood, 2006). Some female computing students said that they consider men as role models, e.g., their fathers or brothers.

On the one hand, having role models can increase girls' self-confidence significantly when there are people with whom they can personally identify (Stout et al., 2011) (see "lack of fit" in previous section). On the other hand, role models that were perceived as a stereotypical computer scientist (i.e., geeky) have a negative effect, especially on the "girly-girls" (Cheryan et al., 2011; Beltrán et al., 2015). Female role models in STEM fields are important at every stage, e.g., female teachers at the secondary level in mathematics and science, female students and faculty members in higher education, and women working in STEM fields (UNESCO Asia-Pacific, 2017). It is essential to make clear what the tech sector is all about, what computing work looks like, and what computer scientists actually do in their job (Stout and Camp, 2014). The absence of role models is of course an effect of the leaky pipeline. Not only do few women study in CS fields, a lot of them also retire after some years for a variety of reasons (work/life balance, lack of flexible working options, etc.) (Good et al., 2012). In addition, it is important that role models are "real". It is not necessary that a role model is at the top of her career to make an effective role model.

> "It is not so important whether the father or the brother has a technical or natural science profession [...] it is about a general and supportive attitude of parents and important people in women's lives that encourage them to follow their own paths. [...] The problem is: The role models are usually super great. An average good student reads an article about a top physicist, who has won several prizes and worked abroad and in addition, has three children [...] that is more daunting. [...] and reinforces the assumption: If you are super good you can make it, even in a field that is not typical for your gender. [...] at University of Technology in Vienna, we have a broader range of different skill levels among male students because there are fewer gender-stereotypical obstacles for them."
> Martia Haas

Mentorship programs can provide effective two-way communication, e.g., for mentees to promote interest and for mentors to reshape thoughts. In addition, mentors help students to develop skills and confidence and can play a key role in generating or maintaining adolescent interest for CS. Some concepts exist to encourage female high school students to volunteer for mentoring programs to help girls from the middle school in CS (Clarke-Midura et al., 2016) with the goal to reinforce both age groups. Other researchers (Cohoon, 2011) use high school aged mentors in CS summer camps and after school programs for primary school girls. Furthermore, other options are informal mentors, like parents (Beam et al., 2002), peer mentoring (Holliday and Luginbuhl, 2004), or formal mentors at the workplace (e.g., for onboarding) or in university for programming novices (Wood and Mayo-Wilson, 2012). A research study (Ko and Davis, 2017) investigates if mentorship enhances students' interest and engagement level in CS courses. The results showed that students who had mentoring relationships (informal or formal) are more interested in computing and engaged significantly more in CS education. Other forms of mentoring include on-site or remote mentoring or long-term and short-term mentoring (Townsend, 2002). Examples for mentoring programs are hack.pledge[37], computerMen-

[37] Mentoring program hack.pledge: https://hackpledge.org/

tors[38], coderdojo[39], or Coding for Kids in Mentor [40]. The TU Graz offers a mentorship [41] as well and a new project from the author of this thesis with the name “RemoteMentor” is at the time of writing at the beginning stages of development[42].

> "Mentoring is among the most effective ways to bring about change, because it reaches across institutions, fields, and even generations. Mentoring unifies women, validates their experience, provides inside information about the system’s workings, trains groups of individuals to challenge the system, and provides the basis for generations of outsiders to both enter the system and, as insiders, demand needed change.”
> (The Association for Women in Science/AWIS, 2017)

The direct encouragement by teachers or role models helps female to attain a certain education level (Gabay-Egozi et al., 2015). Teachers’ guidance and advice are major influences over students’ educational incentives and attitudes. A study conducted by Microsoft (Microsoft, 2017) with 11,500 European women between 11 and 30 years old shows that more than half (57%) of them said that they had a teacher who encouraged them to pursue a STEM career. 60% said they would feel more confident in pursuing a career in STEM areas if they knew that men and women are equally employed in these occupations. Furthermore, role models from history (see Section 2.3) or role models that appear in books, articles, or websites can inspire young women to enter the CS field as well (Townsend, 2002; Back et al., 2011).

To conclude, if girls are exposed to strong female leaders, they rarely express automatic stereotypical thoughts (Dasgupta and Asagari, 2004) and steady contact with successful women (e.g., professors at college) leads to higher career ambitions and stronger implicit self-images (Asgari et al., 2010; Heilman, 2012).

2.4.1.4. Girl’s deficits in experience and motivation.

Interest in STEM areas starts with first performances in mathematics and science (USAID, 2008). While younger students (age < 15) do not yet show significant disinterest in computer science related topics, a number of studies indicate that the majority of teenaged girls rapidly drop out of IT related courses during high school (Zagami et al., 2015). Especially in grades eight and nine, female participation or their interests begin to decline toward STEM subjects (Beyer et al., 2003). Thus, many studies conclude that high school is the key to understanding the field-major segregation in higher education (Gabay-Egozi et al., 2015; Unfried et al., 2015; Mann and Diprete, 2013; Sadler et al., 2012). In the higher grades, male students express more interests towards physics and engineering, while girls were more likely to prefer biology (Gabay-Egozi et al., 2015), medicine, and health (Sadler et al., 2012). These different educational preferences reflect in gender-typical educational choices and embeds gender role socialization. For instance, girls in Ireland are most interested in STEM subjects around the age of 11, but then they are less interested at the age of 15, (n=1,000, 11-18 years old) (RTE, 2016). A three-year STEM initiative in 43 school districts in the US (Unfried et al., 2015) investigated the distributions of career interests between male and female students. Results showed that female students’ interest in STEM career pathways declines among older students,

[38] Mentoring program: http://computermentors.org/kidscode/
[39] Coder dojo: https://www.coderdojoparramatta.org/
[40] Mentoring program: https://locations.sylvanlearning.com/us/mentor-oh/coding-for-kids
[41] Mentoring program TU Graz: https://www.tugraz.at/studium/lehre-an-der-tu-graz/studierenden-mentoring/
[42] Project RemoteMentor: https://www.netidee.at/remotementor

with the sharpest drops between elementary and middle schools. However, the results demonstrate very low interest in engineering careers in girls at all (only careers in energy were of lower interests). Finally, a cohort study on more than 6,000 students showed that the interest changes dramatically in female students whereas the percentage of males interested in STEM appears stable (Sadler et al., 2012). The percentage of female students reporting an interest in STEM fell from 12.1% to nearly half at 7.6% at the end of the high school. Three quarters of the male students already said from the beginning that they are interested in STEM. This initial interest is nevertheless the best indicator for having career interest in STEM when leaving high school. In addition, good grades in mathematics are also suggested as a good indicator and it seems to be a promising chance for interested female students already in primary and middle school levels. However, it reveals that female students in Europe at the age of 11 become interested in STEM subjects and are less likely to express interest in STEM fields when they are at the age of 15. Possible reasons that affects girls' interests can be traced again to girls' negative stereotypes and lower expectations of success (Master et al., 2016). Interest is an important motivational variable because it can affect learning and performance outcomes (Hidi and Harackiewiczy, 2000).

> "It start in the family through different toys and interests, assign to different activities. There are studies that toys for girls are typically not creative, toys for boys are more learning tools, putting parts together, build something. That is one reason that girls are not from the early on encouraged to think in a way that is necessary for technology. There is absolutely no data available that it would be something natural or something in the biology. There is absolutely no scientific evidence on that — that's one thing that's certain. Another reasons would be influences. [...] around the age of 12, well they begin to negotiate their gender identity more but also feel certain pressures [...] sort of peer pressure to go into particular professions and interests. [...] small things can have a huge impact or also offers for boys and for girls, culture imagery, all of that. Some of the basic questions go back to the early 70s." Libora Oates-Indruchová

A study that illustrated students' interests at the Carnegie Mellon University towards CS from 1999 to 2012 observed the following main interests across the genders (Frieze and Quesenberry, 2015): problem solving, building or creating something, working with useful applications, and long-term opportunities. These answers are from participants who are already at university level whereas teenagers mostly put CS on the same level with programming. Furthermore, the female students in this study revealed the importance of working in groups when "times get hard through project work" — thus it is more fun and that plays an important role in the community building as well. The female students at Mellon University in general: enjoy what they do, believe it makes them feel special, and some note that to be a woman in CS is not an issue or others said that there were too few women. However, they did not find a gender difference in attitudes towards programming. Another study (Giannakos et al., 2014), examines the effect of enjoyment, happiness, and anxiety in girls during creative development activities. Findings showed that happiness had a positive effect, anxiety had a negative effect, and enjoyment had a neutral effect on students' intentions to participate in coding activities in the future. Happiness could be increased, e.g., by using humor, fun, and positive feedback, and anxiety could be decreased by praising their development skills and using collaborative learning environments (see next section).

Finally, a research study (Carter, 2006) on students' experiments which included 836 high-school students (363 boys, 423 girls, 50 declined to say) found out that boys were more likely to had prior experiences (40% boys, 27% girls) and 26% of the males and 17% of the females were able to provide a description of the CS major (answers include: programming with around 100 answers, and networking, advanced use, repair, how computers work, computer stuff, good understanding with around 20 answers, and most had "no idea" with around 650 answers = 80%). Even if students had no idea of what computer scientists actually do or what programming is, 11% mentioned that they stay away from CS as major because of programming. Other answers (mentioned in a similar amount in both genders) included sitting in front of the computer all day, and that they would prefer a more people-oriented major. Most interesting — which also can be seen in Figure 2.17 with Biomedical Enginieering — is the positive influence of combining CS with another field of interest such as business or medicine (rated as the number one positive influence in girls). However, computer games or previous experience were also mentioned as positive influences for choosing CS as a major (more by boys than girls).

2.4.1.5. Girl's differences in programming and gaming behavior (the male-oriented game industry).

Only a small amount of research exists that shows differences in programming between boys and girls. Studies which focused on feature discovery show that male users adopt higher amounts and different types of features compared to female users (Beckwith et al., 2006; Beckwith and Burnett, 2004). Female students were significantly slower in trying out new features and were significantly less likely to use them again, whereas males try less familiar features early. Fewer uses of features correlated with a low self-efficacy only in girls. The study also refers to differences in the tinkering behavior and it states that male students seem to benefit more from it. However, tinkering also helped females to gain valuable information about the features and increased their self-efficacy. Low tinkering interactions and low self-efficacy occurs in girls if they use environments that are too complex. The study concludes that gender differences exist in the way students solve problems, which may indicate a need for a more supportive feature designs. This is also coherent with other literature (Grigoreanu et al., 2008) and often more related to testing and debugging activities.

In Section 2.4.2.4, the author presents that more and more gamers are female, and that there is already a small shift in putting more focus on gender inequality issues in games (Jenson et al., 2007; Williams et al., 2009; Martin and Rafalow, 2015; Paaßen et al., 2017). However, most developed games only appeal to a male-dominated audience and exclude female gamers, there are less successful games available for girls as there are for boys (Google Inc., 2018), most games do not appeal to a female audience, and there exists less motivation for girls to become gamers (only 30% consider themselves to be gamers (NewZoo, 2017)). In addition, most boys say that they prefer to play video games with boys (Jenson et al., 2007). The gender bias has a negative impact on women who play (Ashcraft et al., 2016; Shaer et al., 2017) and gender stereotyping in games can be damaging (e.g., a princess who needs a male hero to rescue her may serve as a trophy). A closer look on existing video games shows that nowadays, more and more strong women occupy the main role in video games. Examples from 2017 are a redesign of the famous character Lara Croft[43], Tomb Raider, who became a much more realistic young woman in her most recent incarnations (see Figure 2.20), the new hero Aloy[44] from "Horizon Zero Dawn", an outsider with an interesting story, or "The Scythian" from "Sword and Sorcery"[45], a video game for mobile platforms (she demonstrates that women can be heroes just as well).

[43]Tomb Raider: https://www.tombraider.com
[44]Horizon Zero: https://www.guerrilla-games.com/play/horizon
[45]Sword and Sorcery: https://play.google.com/store/apps/details?id=com.capybaragames.sworcery

Figure 2.20.: Revolution of Tomb Raider character Lara Croft (from left to right, starting with the original game) (Tomb Rider, 2017)

Nevertheless, video games with female protagonists are still in the minority (Jayanth, 2014). A survey that was conducted with 1,266 gamers shows that 75% of the female gamers rated female protagonists as "very" or "extremely" important (Yee, 2017b). This was more than three times higher than the number of answers from male gamers. However, most characters in games are male and not inclusive at all, for instance, Grand Theft Auto [46] (which shows sexual harassment and killing of female characters), female characters with less clothing, like Juliet Starling from Lollipop Chainsaw [47], or other examples (e.g., Tony Hawk's Underground, Halo, and Mortal Kombat). A study shows that the oversexualization of women in games peaked in 1995 and then declined (Lynch et al., 2016), but women still occupy secondary roles and are objectified more often for several reasons (like marketing, lack of female game developers, or lack of game developers who are aware of gender issues and studies) (Kondrat, 2015). Game developers have to rethink their strategies now that games are getting more appealing to a wider audience in order to target all cultures, genders, and interests. In addition, developers should put more emphasis on the personalities, emotions, and backstories of characters in the games.

Video, computer, and mobile games are, of course, entertaining and fun. However, it is important to know that there exists a correlation between students who do not play video computer games and those who describe their computer skills as insufficient or do not spend much time on technical devices (Davies et al., 2014). In general, older teenagers (15 to 17 year-olds) are less likely than younger teenagers (12 to 14 year-olds) to play computer games on a daily basis (Lenhart et al., 2008). In addition, students interviewed mentioned that the majority of their computer science knowledge was acquired at home or with friends while playing or in social networks (Goode et al., 2006). Another study shows that only 1% of the finalists from Austria at the International Computer Science Olympiad[48] for teenage high school students over the last 12 years were girls. A closer look at the

[46]Grand Theft Auto: https://dotesports.com/the-op/news/grand-theft-auto-vi-news-21158

[47]Lollipop Chainsaw: https://aliciabear77.files.wordpress.com/2013/06/1eab9ecceb014ae0b33c87135641bed1.jpg

[48]Internation omputer Science Olympiad: http://stats.ioinformatics.org/results/2016

numbers of women in the game development sector in general shows the same low number. This is especially interesting because the number of women playing games has slowly increased (see also Section 2.4.1.5). According to a study commissioned by the Internet Advertising Bureau in the UK, women account for 52% of the gaming audience (Jayanth, 2014). However, in Britain, only 12% of the professional game designers and only 3% of programmers are women (Jayanth, 2014).

Before focusing on strategies to reinforce female teenagers, this section summarized actions and attitudes which discourage young women. Figure 2.21 points out all negative influences from the previous section, which resulted in a picture of a negative stereotype of "Girl in Tech".

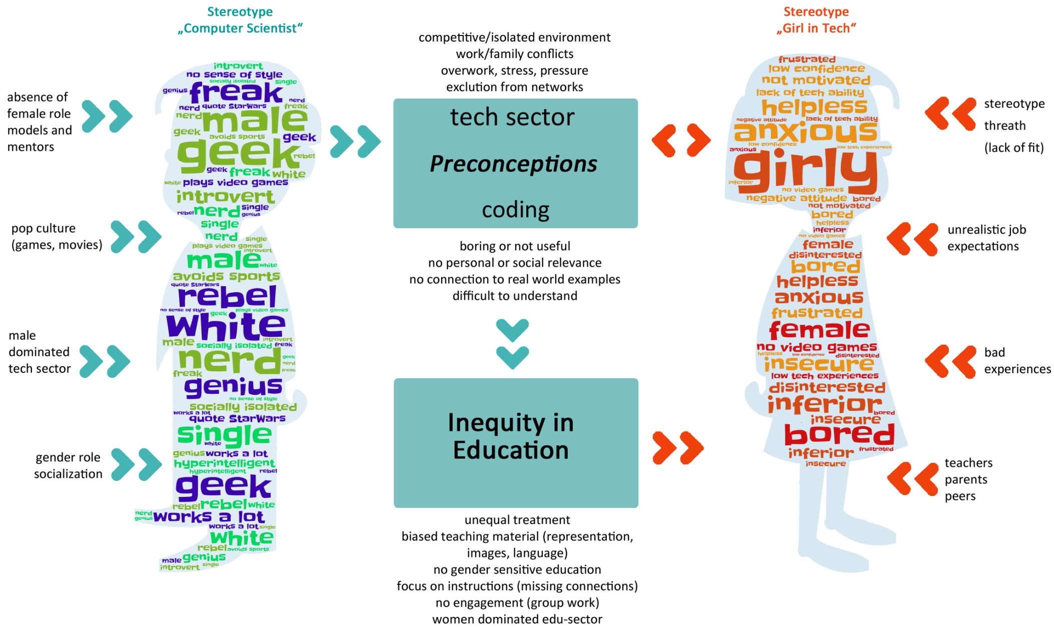


Figure 2.21.: Shaping of the negative stereotype of a "Girl in Tech" as a result of the stereotype "Computer Scientist", preconceptions and inequity in education.

It is worth mentioning that of course many of the men also did not appear to fit the narrow computer scientist geeky CS stereotype (Frieze and Quesenberry, 2015; Margolis et al., 2014). Despite this, the picture is relevant about commonly held beliefs and how they influence young girls' interest. Attributes assigned to the stereotype "computer scientist" include geek, male, white, nerd, rebel, freak, socially isolated, plays video games, genius, introvert, obsessed with working, single, quotes StarWars and/or StarTrek, avoids sports, has no sense of style, and is hyper-intelligent. The author of this thesis created as a result of the literature review the stereotype for "Girl in Tech" which is formed by the male stereotype, the preconception of the tech sector and coding, and inequality in education. This stereotype has the following characterization: not motivated, plays no video games, disinterested, lack of tech ability, low confidence, negative attitude and low experience towards tech, helpless, anxious, bored, inferior, insecure, frustrated, and of course "girly". The case studies from Frieze and Quesenberry (Frieze and Quesenberry, 2015) show that this does not reflect the opinion from their CS students at Carnegie Mellon University, which describe the typical CS student more positively, e.g., creative, passionate, focused, smart, or diverse, which seems to be more the case. The students themselves play an important part as well in shaping the CS culture and environment from the inside.

To conclude, if the CS environment is not something were females are happy to study and work in, the environment has to change itself (Dee et al., 2009).

2.4.2. Creating suitable learning environments to reinforce female teenagers.

Framing a supportive classroom setting is a critical factor and literature suggests that interventions should specifically target the classroom climate to strengthen teenage girls' confidence to motivate female students extrinsically as well (Haines, 2004). Positive first experiences in coding for girls may direct their future career choices towards STEM fields. Many of the changes in teaching and learning that resulted from the study of the empowerment of girls improved the situation for all students, not just for girls (Kafai, 2008). The author assumes that learners of all genders who become game designers and creators of their own learning content through a constructionist learning environment will significantly contribute to closing the divide and participation gap in digital culture.

This section first provides suggestions from the literature that seem to be promising in reinforcement of female teenagers in coding classes, with focus on both the classroom settings and girls' playing behavior. The author then tries to answer the question of if this target group needs customized tools in order to feel engaged. Finally, the author focuses on different "Girls-only" initiatives worldwide and nationally by presenting best-practice examples. These factors which should create inclusive classroom environments have been already discussed in the author's paper (Spieler, 2018; Petri et al., 2015).

2.4.2.1. Problem-Based Learning (PBL) environments.

Some researchers (Monitise Group Limited, 2014; Carter, 2006) argue that girls are more interested in relations between ICT and other fields, e.g. computer scientists are involved in many disciplines, e.g., bioinformatics, medicine, environmental modeling, or media experts. Other research (Cukier et al., 2002) supports the argument that it is important to consider alternative routes to IT because in reality it is not all technical; there are multiple points of entry (design, analytics, problem-solving skills, etc.). Otherwise, mathematical or logical abilities are more associated with this field, rather than literacy or interpersonal skills. The geeky, nerdy, isolated, fanatical computer expert who represents the stereotype of computer science is certainly not something most young women strive to become (Cheryan et al., 2013) (see also previous section). The technology sector needs people who can design, develop, analyze, and manage information technology rather than people who are mainly developers who do objective assessment of technology or do just programming stuff (Cukier et al., 2002).

A problem-based learning environment allows for the exploration of different contents to achieve related objectives. Thus, the contents are not isolated but linked with a common goal of coding a game (Paderewski et al., 2015). As described in previous sections (see Section 2.3.2), computer science education in lower grades most likely consists of learning about Office products or media presentation tools. Researchers point out the importance to directly link the subject of computer science with professional IT jobs to positively influence girls' interest for IT related fields (Weibert et al., 2012). In some CS courses, low engagement levels are the default norm. These courses focus not on problem-solving skills or creative skills but on basic input and output processes (see Behaviorism theory, Section 2.1.3). Thus, the CS classes mainly rely on given samples in textbooks, with whole lines of code, and following directions without using logical reasoning. This setting is not only less challenging for all students, but also boring, and it even prevents them from truly understanding the programming language and the concept itself. This concept does not show the multiple ways they could use for solving the problem, does not foster any form of group work or class discussions, and finally, leads to negative experiences (for all genders).

> "We do know also from other areas that women and girls prefer different things. [...] Problem tasks — verbally — I think that stimulate imagination and creativity because I can imagine something. Something more practical that I am doing with the formulas. I found this incredibly exciting. [...] A different way to relating to the subject. I think the narrative is one of the issues. " Libora Oates-Indruchová

To conclude, it is important to present computer science as a variety of perspectives and possibilities (Khan and Luxton-Reilly, 2016) and to make connections to other subjects (Fisher and Margolis, 2002). As already described in Section 2.1.1 and 2.3.1, the Federal Ministry of Education should focus more on integrating ICT in the regular curriculum, thus focusing more on interdisciplinary lessons that apply ICT. This works by shifting the aim and seeing computer science as something that helps to make the world better by creating programs and products or new ideas, which definitely contrasts with many of the existing negative stereotypes.

2.4.2.2. Collaborative Learning (CTL) environments.

Collaborative learning activities allow communication and mutual decision-making (Denner et al., 2005; Chance and Bowe, 2015). A collaborative learning structure has many benefits. For instance, students who help each other address complex problems work more efficiently (Barber et al., 2005). In addition, the assignment of different roles and responsibilities within the team is a central peer component. Collaborative group work, especially between girls, fosters their sense of community and self-confidence (Chase and Okie, 1999; Pollock et al., 2004). To foster collaborative skills and practices, it is important to not only divide the tasks, but also to work together and share ideas, thus supporting collaborative environment.

According to a study, adult groups which consist of women only outperform groups which consist of male members in a collaborative task (Chase and Okie, 1999). According to another study (Beyer et al., 2003), the most important factors for women to choose CS as a major are 1) positive interactions with CS and 2) collaborative programming projects. In contrast, men were more likely to mention 1) prior experiences and 2) workload to be most important. The gender composition of groups plays an important role, since girls' pairs shared information more readily and used problem-solving and verbal strategies more often (Barker et al., 2014). Successful group work means being able to understand different opinions of group members and discussing the problem and ideas, which seems to be easier within girls' groups (Gibbs and Mueller, 1990). Boys, in contrast, are more likely to talk about disagreements rather than focus on the idea or utilizing certain strategies. A study (Tsan et al., 2016) showed significant differences in the quality of artifacts produced by groups of girls and boys in regard to the PECT model (Progression of Early Computational Thinking). Groups of female students performed significantly lower than other groups (boys and mixed groups). On the one hand, male students appear to benefit more from heterogender groups by, developing their leadership skills and increasing their self-efficacy in problem-solving. For female students these kinds of pairings can reinforce gender stereotypes (Lockheed and Harris, 1984). This raises the question of if female students should be "forced" to be in groups together, which can lead again to a stereotype threat scenario. The author suggests letting students decide for themselves.

The COR model (Gibbons, 2013) refers to “Choice of Activities”, “Option with Others” and “Revision” and should help to manage three root causes of anxiety which reduce students’ motivation in game design. These are: having no choice, being dependent on the performance of others (group work), and having only a single chance to succeed. “Choice of Activities” or a “Freedom of Choice” in game design provide a setting in which students can choose their own game design, MDAs, and characters (see Section 2.2.1) freely and to their own preferences and a freedom of choice in group work. Competition can cause anxiety and may discourage female students (Hercy et al., 2009).

Furthermore, the literature supports the argument that classroom interventions for girls should focus on providing guidance and tutorials (Carter and Jenkins, 1999). Female students often find small group tutorials more useful than male students find them, and enjoy them because they accomplish small steps while they get one step closer to the finished program. Thus, a scaffolded game design is seen as an effective approach for female students to develop fundamental computing concepts by completing almost finished games (Games and Kane, 2011). In contrast boys, are more likely to play around, they are seen as more competitive, do not follow the rules exactly, and like to manipulate tools in a more exploratory way (Jones et al., 2000).

2.4.2.3. Building creative environments.

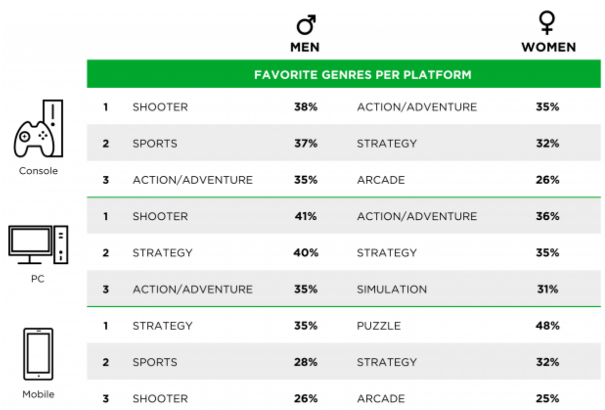
For women, creativity and interest in STEM professions are often related (Microsoft, 2017), but there is a lack of practical relevance in a lot of these subjects. A European-wide study in which 11,500 young women between 11 and 30 were interviewed showed that girls between the ages of 12 and 16 are the most creative. Thus, female teenagers at this age should be supported especially in skill trainings and STEM. Approximately every third women that has been asked (33%) criticized how scientific topics were explained in schools and especially the topics in computer science. Within the study, it is mentioned that those subjects are taught from a more "male perspective". The European educational researchers state that creative teenage girls are particularly interested in STEM subjects and could be inspired by the world of programming with creative initiatives. It is therefore important to eliminate the prejudice that STEM professions are not creative. Through good game design, creative environments, and customization, a broad spectrum of girls can be reached (Subsol, 2005). Another research study argued that facilitators should promote especially creative IT careers that are more driven by creative thinking and design: for instance, computer animation, game design and web design (Wong and Kemp, 2017). As a result, more female students expressed their interest in IT careers.

With experiments and hands-on activities in the classroom, different areas can be presented, thus enabling a good start into the digital future for girls (Trauth et al., 2004). The game design process for girls in general should allow the involvement of identity exploration by transforming games to suit personal preferences. If these design tasks allow room for exploration, customization, and creativity at the same time, female students feel engaged. In addition, a starting point for participation and collaboration is provided. According to a study that elaborates game design and development for middle and high schools girls, such courses increased students’ self-efficacy, helped them in acquiring design and coding skills, and engaged them in the activity (El-Nasr et al., 2007).

2.4.2.4. Let play, create, and design.

Teaching good game design and development skills (see Section 2.2.1.1) is especially important for girls because girls are not that likely to play games, play less active roles, and therefore they are less endowed with problem solving skills (Bellotti et al., 2011). In addition, literature argues that even if the number of female students who play video games increased in recent years, male students have a greater interest in playing games (Jenson et al., 2007). GPL theories (see Section 2.2.2) strive to motivate students by playing games. Thus, in this section the author of this thesis first provides an overview about girls' playing behaviors (video games and mobile games) to get a sense of what games are played by women in general and to get a clearer picture of the target audience. Second, the author summarizes typical design patterns girls tend to use in creating their own games.

Mobile games are very popular and the market is increasing as already described in Section 2.1.4.2. A recent study of Google emphasizes on females with its campaign slogan of "Change the Game" (Google Inc., 2018). Google, as one of the biggest app store providers, partnered with the gaming intelligence provider Newzoo[49] to examine American female players' experiences and perceptions by performing a quantitative study (3,330 female participants aged 10-65) (Google Inc., 2018). According to this study, 65% of women play mobile games, compared to 2011 when only 31% of women played mobile games. Furthermore, half of mobile gamers are women and 64% of women prefer smartphones to other platforms (38% of the men do so). A percentage of 43% of women play mobile games five times per week or more (38% of men to so) and they prefer to play at home, in the bathroom or in bed. Mentioned reasons why women play are restful moment, stress relief, or entertainment, and 60% say that they feel good while playing. In contrast to men, who play three or more different kind of genres, women play two or fewer different genres, they communicate less about play with friends (men 44%, women 27%), and only 33% of women invest in mobile gaming (52% of the men do so). This study also mentions that even their Google Play Store features more male identified characters than female (44% more male characters than female in the 100 most revenue-generating games). In addition, only 30% of the women who play feel that the games are made for them. Finally, only 29% of the women who play games identify themselves as gamers (57% of the men do so). Google Play believes that mobile gaming can inspire creativity by showing users new worlds, and let them engage. The company sees a huge opportunity to make mobile games more diverse, inclusive, and engaging for all players.



Platform	#	MEN	%	WOMEN	%
FAVORITE GENRES PER PLATFORM					
Console	1	SHOOTER	38%	ACTION/ADVENTURE	35%
	2	SPORTS	37%	STRATEGY	32%
	3	ACTION/ADVENTURE	35%	ARCADE	26%
PC	1	SHOOTER	41%	ACTION/ADVENTURE	36%
	2	STRATEGY	40%	STRATEGY	35%
	3	ACTION/ADVENTURE	35%	SIMULATION	31%
Mobile	1	STRATEGY	35%	PUZZLE	48%
	2	SPORTS	28%	STRATEGY	32%
	3	SHOOTER	26%	ARCADE	25%

Figure 2.22.: Statistics from (NewZoo, 2017) show women prefer mostly action/adventure genres.

In comparison, the company NewZoo published current numbers of the video game industry (NewZoo, 2017), showing that 46% of gamers across these 13 countries are women (aged between 10-65) who play on different consoles (35% PC, 48% mobile, 23% console). A percentage of 12% of women who play games are 10-20 years old. Female players hear about new games from friends/family (39%, men: 27%), social networks (20%, men 17%) or reviews/game sites and advertisements (18%, men 24-26%). Figure 2.22 shows preferred genres per platform and gender.

Another statistic rated family/farm simulation and match 3 games as the most preferred for female gamers (69% who are playing these genres are female) (Yee, 2017a). Match 3 games can be sum-

[49]Newzoo: https://newzoo.com/

marized as puzzle games where you mostly need to combine three tiles together, for example Candy Crush Saga[50] (Julkunen, 2015). These statistics emphasize that the genre averages range from 2% to 70%, thus developers should never focus on general statistics that consider all genders.

Amanda Ochsner (Ochsner, 2015) collected design patterns from different girls initiatives from 1990-2007. These and literature from the same time (Agosto, 2003; Gorriz and Medina, 2000; Heeter et al., 2000) of typical design patterns, characteristics, and content girls tend to like are summarized in Table 2.3.

Table 2.3.: Design, characteristics, and content among girl games.

Design	Game characteristics	Content
• exploration • collaboration • challenge • vicarious adventures • sophisticated graphics and sound design • role playing • realistic design	• rich narrative • roles involving positive action • appropriate levels of challenges • opportunities to design or create • engaging characters • communication and collaboration use of strategies and skills	• storylines and character development • real-life locales • characters who are in charge of decisions and actions • to create rather than to destroy • involving simulation and identity play • chance to swap identities

The programming environment Scratch (see Section 2.1.4.1) leverages game design and makes coding more accessible for a broader user group, especially novice programmers (Fields et al., 2014). A research study which observed game designs in Scratch (with a focus on racial and ethnic diversity) shows the following (Richard and Kafai., 2016): In general, most projects from female game designers focus on popular TV shows, games, or toys, or refer to mazes, dragons, and other pop culture creations and stories. Furthermore, an inspection of 52 Scratch games showed that female students used the most interactive objects that operated through mouse clicks, keystrokes, animations, stories, and projects that included multiple genres, e.g., music with interactive objects (Kafai et al., 2012).

2.4.2.5. Customized services for female teenagers.

Coding initiatives designed especially for females can support young women in their decision to choose a computer science career. It is important that these initiatives are promoted the right way and out of comprehensible reasons to prevent further strengthening of stereotypes (Engler and Wieland, 1995; Spivak, 1990). As has already described in the last sections, there is a high probability that girls being assigned a stereotype-threat have more negative thoughts regarding their abilities caused by their fear of confirming the negative stereotype (Cadinu et al., 2005). However, girls' initiatives create opportunities to focus on their interests and to enable them to socialize with other girls interested in computer science (Alvarado et al., 2017). In this section, first, the need and criticism of tools customized especially for girls is discussed and second, successful international and national girls-only initiatives are presented.

Some may argue against developing tools/products specifically designed for women. On the one hand, many examples exist where companies have marketed to women as some special subclass of

[50] Candy Crush Saga: https://play.google.com/store/apps/details?id=com.king.candycrushsagahl=en

humanity with totally unique needs and an unshakable preference for the color pink (Atlanta Parent Editorial, 2017; Rommes et al., 2011). Some products are of course unisex, e.g., food, handicraft tools, or laptops but that does not reflect the opinion of all. There are examples from companies that provide, pink handicraft tools (which costs even more than the standard tool). Computers for women are an issue as well, e.g., "Petticoat 5"[51] from the 80s (with larger keys to type with manicure fingernails) or "Floral Kiss" from today[52]. Such products are so called "pink technologies" created to increase women's consumption (Rommes et al., 2011). Researchers argue that children have no choice but to follow these socially stereotypical norms.

On the other hand, it is necessary to see the needs of women, e.g., in speech recognition systems (Baustädter, 2016) or in games. In addition, some companies also focused on inclusive educational software (e.g., Girl Games[53], Purple Moon (Gorriz and Medina, 2000)). Researchers emphasize the importance of developing games with female protagonists and promoting games especially for girls (see Section 2.4.1.5 and Section 2.4.2.4). According to Google (Google Inc., 2018), to create inclusive games a developer should 1) know the audience (current and potential), 2) look at the game designs and which may exclude potential players (imagery, iconography, etc.) and test it by store listing experiments, and 3) have a diverse team (range of different perspectives) (Martinson, 2005). In addition, it is important to pay attention to the store icon, screenshots, videos, and launching of female characters. This will be discussed in more detail in the result section 5.3.

Another discussion is about inclusion of gender perspectives in programming environments and software (Burnett et al., 2010; Ramnarine-Rieks, 2012; Huff, 2002) or feature discovery (Grigoreanu et al., 2008; Beckwith et al., 2006; Beckwith and Burnett, 2004). Researchers argue that most environments are designed by men and thus, women have to adopt a male perspective when working with technologies (Ramnarine-Rieks, 2012). As discussed in previous sections, male-dominated environments have a negative impact on women's choices and reinforce a mainstream culture, where inclusion of gender is a serious problem. Thus, an inclusion of gender perspectives during the planning and development phase is important, as is considering the impact of gender relationships with the product. Evidence exists that reasons for the gender gap are also related to the software environments in the realm of end-user programming (Grigoreanu et al., 2008; Burnett et al., 2010). In addition, it is stated that inclusive feature discovery has no disadvantages for men (Grigoreanu et al., 2008). Furthermore, it is recommended to include expert reviews or user testing, which can help to eliminate computer interaction flaws in software (Huff, 2002; Williams, 2014). The GERD model (Gender Extended Research Development) (Claude et al., 2014; Maaß, 2018) proposes the following approach when integrating gender in computer science (tools): define impulses (current topics, situations, surroundings), initiations (initial position, target group, predicted outcomes, methods), analysis (user & context, technology, risks), define a model/concept (theory, system-specification), realization (prototype, development, design, implementation), evaluation (verification, test, user concept/perspective), and dissemination (documentation, maintenance, support, publications, go into action). Factors that are considered in GERD are knowledge, power structures, usefulness, working culture, the human image, language, relevance, and values.

> "Personalized tools for girls are a very critical issue. I think it is assumed that there is a homogeneous gender group of girls while other forms of difference are set in the background and it only counts whether you are a girl or a boy [. . .] girls are stereotyped

[51] Petticoat: https://www.chonday.com/15509/marketing-computers-for-women-in-the-80s/

[52] http://www.businessinsider.com/fujitsus-floral-kiss-computer-women-2012-10?IR=T

[53] Girls Games: http://www.wired.com/wired/archive/5.04/es$_{g}$*irlgames.html*

> on the spot and assigned certain attributes, e.g., they are not interested in technology, they may be less tech-savvy, they need a different color [. . .] so the stereotyping is carried on and on.” Marita Haas
>
> “If you offer them pink, and only pink — they will go for pink. [. . .] a customized version [. . .] where the user has the possibility to customize rather then you theme them a particular version. [. . .] because you assign them already to a stereotype and maybe not doing them a good service because if the end aim is to get them motivated for programming then they will be unfamiliar with how these tools look like in the real world.” Libora Oates-Indruchová

In Section 2.4.2.5, the author presented a range of external initiatives and organizations which offer playful coding activities off-school like coding clubs. These coding club initiatives have predominantly male participation (Zagami et al., 2015). To promote initiatives for female teenagers as girls only, e.g., coding clubs or summer camps is therefore important in many ways (see Section 2.4.1). These initiatives serve as vehicles to interest girls more deeply in ICT, to foster their sense of belonging and self-efficacy, and to show them new opportunities in their lives. If such activities are promoted in schools, teachers have the conflict to provide similar activities for boys as well. Moreover, situations where females are preferred to males can lead again to a range of negative impacts (stereotypes, threats, discrimination, etc.). Thus, many external initiatives exist that promote girls-only courses. Overall, the key benefits of girls-only initiatives are social encouragement (reinforcement of CS by peers, mentors, role models), self-perception (interest in problem solving, creative and collaborative environments), and career perception (job clarity, personal relevance) (Zagami et al., 2015). Furthermore, facilitators and teachers report the difficulty to engage girls and boys equally in traditionally male-dominated subjects such as computing. Thus, coding initiatives for girls may improve women’s participation in such activities. The following tables present successful girls-only initiatives worldwide, in Europe, and in Austria (see Table 2.4)[54], and compare them by age, target group, course content, and setting.

[54]Girls in Tech: http://girlsintech.org/; Girls Make Games: http://girlsmakegames.com/, Girls make Games: http://girlsmakegames.com/, Django Girls: https://djangogirls.org/ and https://djangogirls.org/vienna/, Made with Code: https://www.madewithcode.com/, Girls develop IT: https://www.girldevelopit.com/, Girls who Code: http://girlswhocode.com/, Black Girls Code: http://www.blackgirlscode.com/, Seattle Girls School: http://www.seattlegirlsschool.org, Wise: https://www.wisecampaign.org.uk/, Tech Women: https://www.techwomen.org/, Indian Girls code: http://robotixedu.com/indian-girls-code/, Girls’n’Code: http://girlsncode.com, CoMaed: https://www.comaed.tugraz.at/index.php/wbindex/start

Table 2.4.: Successful coding courses from all over the world.

Initiative	Details	Age/Country	Content	Setting
Girls in Tech	since 2007 non-profit	women in general global	empowerment, engagement and education network to boost visibility of women in tech jobs	bootcamps, global classrooms, mentorship
GIRLS MAKE GAMES	educational company	8-17 year olds global	web, Angular, Git, HTML, CSS, PHP, MySQL access to knowledge through games	game workshops
django girls	since 2014 non-profit	girls/women global and local facilities	web application using HTML, CSS, Python and Django	workshops, open sourced online tutorials
Made w/ Code	since 2014 by Google	school girls worldwide	a community-based site filled with projects to encourage girls to learn coding,e.g., make a film, build an app	online courses
Girls develop IT gdi	since 2010 non-profit	school girls US	learning web and software development, e.g., HTML/CSS, Javascript, Github/Git	in-person classes and community support
girls who code	since 2012 non-profit	10-11th grade girls US	learn coding and get exposure to tech jobs for beginner to advanced	clubs, campus and summer immersion
black girls CODE	since 2011 company	7-17 year old girls US	introducing coding by robotics, hosting talks from women in tech	workshops after-school programs
Seattle Girls' School	Bill&Melinda Gates Foundation	Middle school Seattle/US	to encourage girls in math and sciences by building robots, and creating new career paths	a girls school
WISE	non-profit	girls/women UK	career development programs for STEM, returners programs, media and presentation skills	training courses for companies, women in STEM, WISE conference

TECHWOMEN	U.S. department of State Bureau	women in Africa, Central / South Asia, Middle East	project-based mentorships, female role models providing access and opportunities	workshops, networking events, mentoring
Indian Girls Code	by Robotix learning solutions	7-12 years old India	coding and robotics education	workshops
girls n' CODE	since 2015 company	for coding novices Vienna/Austria	web design (HTML, CSS), coding, blogging	courses, videos,talks
CoMaed „Computer und Mädchen"	TU Graz	+ 10 years Graz /Austria	robotic, coding, web-design, for beginners and advanced	courses

> "Segregation works for girls in particular because they are not compared to boys, they are not teased, they can communicate with each other. I think it has been confirmed that in some cases the girls only environment can stimulate particularly in when we are talking about a discipline that is traditionally not considered for women. Then I think it is important that they are by themselves. It is important that they can have their own community. [...] When you think you have an environment where competition and hierarchy has such an impact that it doesn't stimulate the girls to a better performance but rather intimidates than that's a good reason to separate them or try to remove the inappropriate influence. This will allow collaboration and different ways of creativity."
> Libora Oates-Indruchová

> "I believe that such initiatives are helpful in order to create a basis, to give the girls self-confidence. For example if they join our 'CoMaed' course and see that they can do it, they also have a different standing in the mixed groups [...] to see: I've tried that once and I can do it! Especially in technology boys have often experiences in tech, since they are more influenced from an early age. And if girls can make those experiences too, they start with a healthy self-confidence." Gudrun Haage

To conclude, many initiatives exits with the common goal to reinforce young women in coding. Such initiatives also exists for other disciplines, e.g., in music[55]. The goal is to provide new frameworks for introducing computer science to broader and more diverse audiences.

2.5. Summary Literature Review

In conclusion, in the literature review chapter fundamental concepts for (playful) coding education have been summarized as well as an overview has been provided about the situation of women in technology. At the beginning of the chapter, the current situation of CS curricula in schools has

[55] Girlsrock: http://www.pinknoise.or.at/ and http://girlsrock.at

been presented. Here the author assumes that coding courses (in Austria) in the future must be more interdisciplinary and foster hands-on (tinkering) experiments through project work. To prepare age appropriate teaching material for teachers is essential to support them with new challenges of the CS curriculum.

The underlying relationship between different learning theories of Constructivism and Constructionism from the past and new concepts from today like CT or the UDL show the importance of problem based and self-directed learning, engagement, and collaboration, as well as constructing new knowledge and sharable artifacts by using new technologies. Intrinsic and extrinsic motivation plays an important role in all learning theories and shows ways to reinforce students for learning by fun and self-efficient activities, which provide interest in STEM, learning goals and frameworks by using a gamified and playful approach. To support coding activities two trends are presented in more details: block based visual oriented coding and mobile learning. Furthermore, terms like games and play have been defined as well as different game based learning strategies, like GBL, gamification and GDBL were presented. Subsequently, a closer look at the game jam approach has been provided.

The next part provided an overview of the broad research area of the gender gap in IT. It shows actual numbers from all over the world and from different fields, which picture the global gender gap, and also show the exclusion of female teenagers from CS university programs, worldwide and in Europe/Austria. This part concludes which the argument that it is important to support women in STEM to improve the situation for all (companies, industries, or in working teams). Subsequently, reason have been provided which indicates that the gender gap already origins in teenagers and is most critical in students in middle school. Female teenagers between 12-15 years decide their future careers and many indicators speak against to decide for a CS career, e.g., stereotypes who have a huge impact, preconceptions, the male tech/gaming industry, absence of female role models/mentors and of course a lack of sense of belonging, summarized in the image of the "Girl in Tech". Finally, several solutions for a gender unbiased classroom have been presented and how to encourage girls with games, inclusive tools and girls only initiatives.

Although gender differences in experiences, self-efficacy, lack of fit, motivations, interests, problem-solving styles, learning styles, and gaming behavior are all implicated in this issue, it is important to remember that no female teenagers is statistically "all-female", which is true also for male. However, like already mentioned, improvements for females have the potential to improve the situation for everyone and support these differences does not penalize either gender — it helps everyone.

The literature review defined the foundation for the practical part of this doctoral thesis. Now that all essential background knowledge have been presented, this sentence marks the end of the theoretical part of this thesis.

Chapter 3

Catrobat and the No One Left Behind (NOLB) Project

The focus in the subsequent chapters lies on the Catrobat project, the educational app Pocket Code, and the corresponding European H2020 project No One Left Behind (NOLB). In this chapter, the whole project is introduced as well as the setting of the Austrian pilot. Finally, the chapter concludes with the implementation of different frameworks, resources and tools during NOLB. A Game-Making teaching framework has been created and in addition, a new app with the name Create@School has been developed for the use at schools. This app is a more tailored version of Pocket Code for academic purpose, with predefined templates for students to start with an almost finished game and accessibility settings have been added for students with special needs. Create@School is also linked to other services such as behavioral tracking to collect events during coding, as well as the Project Management Dashboard (PMD) for project submission and assessment by teachers.

These Catrobat tools and services have been evaluated during the time of the project according to their usability and attractiveness. Their evaluation is part of Chapter 4 and build the foundation for the results of this thesis in Chapter 5.

3.1. The Catrobat Project

The Free Open Source Software (FOSS) non-profit project Catrobat[56] was initiated 2010 in Austria at TU Graz. The multidisciplinary team develops free educational apps for teenagers and programming novices. The aim is to introduce young people to the world of coding. With a playful approach, young people of all genders can be engaged (see Section 2.2) and Section 2.4.2.4) and game development can be promoted with a focus on design and creativity (see Section 2.2.2). Catrobat is the official name of the project's visual programming language as well. The project follows an interdisciplinary approach through worldwide collaborators. More than 500 participants, most of them students of TU Graz, participated in this project. A first public version of our free app published in 2014, with 46 releases of the main coding app as of April 2018. Our app currently has more than 700,000 users in 180 countries, is natively available in 40+ languages (including several languages not directly supported

[56]Catrobat: https://www.catrobat.org/

by the underlying operating system), and has been developed so far by almost 1,000 volunteers from around the world.

3.2. Pocket Code: Creating Personalized Apps

One of the apps of Catrobat is Pocket Code, a visual programming language environment that allows the creation of games, stories, animations, and many types of other apps directly on smartphones or tablets, thereby teaching fundamental programming skills. This app consists of a visual Integrated Development Environment (IDE) and a programming language interpreter for the visual Catrobat Programming language. The IDE automatically translates the underlying code parsed by the XML file into visual brick elements and vice versa. With the use of simple graphic blocks, users can create their own game, colorful animations, or extensive stories directly on the mobile phone without prior knowledge. Visual and block based coding language helps novices with an easy to use interface and predefined Lego® style bricks (see Section 2.1.4.1).

The drag and drop interface provides a variety of bricks that can be joined together to develop fully fledged programs. The app is freely available for Android on Google's Play Store[57] and soon will be available on Apple's iTunes Store for iPhones. The target audience for Pocket Code are teenagers between the age of 11 and 17 years old who have access to or own an Android smartphone. Figure 3.1 shows Pocket Codes' UI and an example program with "Alice in Wonderland" characters.

[57] Pocket Code: https://catrob.at/pc

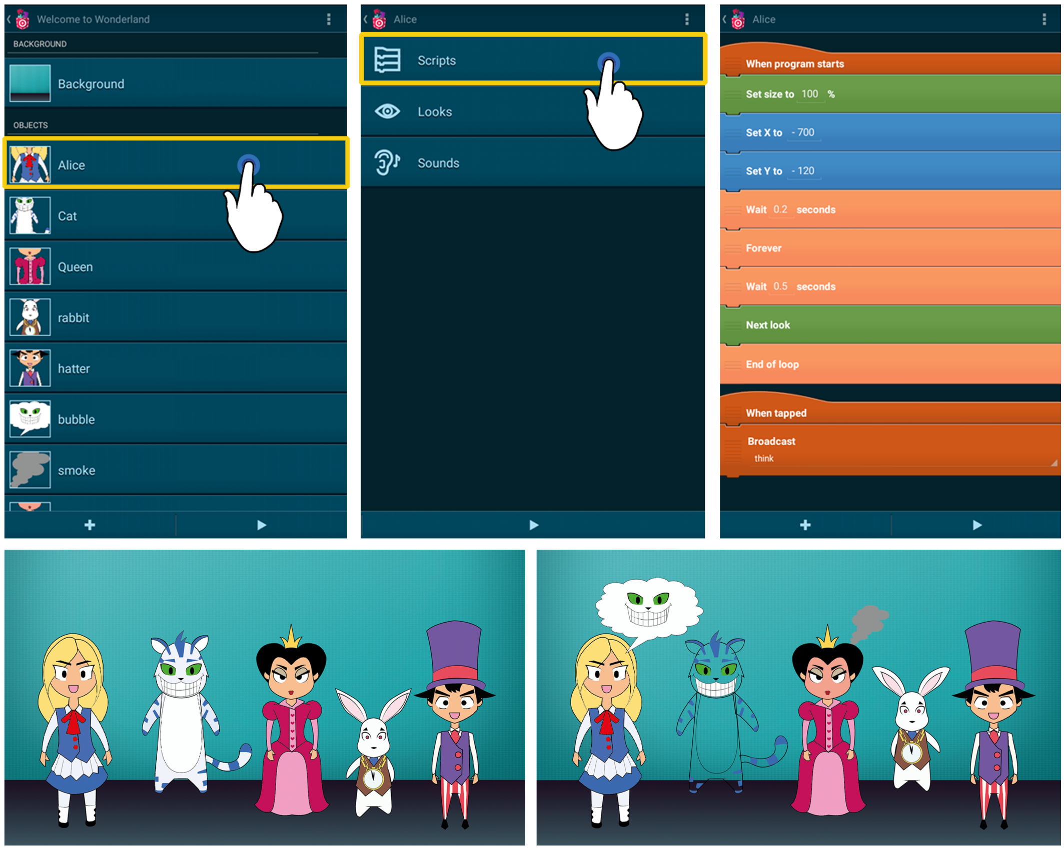


Figure 3.1.: Pocket Code Alice themed program.

3.2.1. Pocket Code: the mobile integrated coding environment.

Programs in Pocket Code follow a similar syntax to the one used in Scratch (see Section 2.1.3.1). Programs in Pocket Code are created by snapping together command bricks as well. They are arranged in scripts that can run in parallel, thereby allowing concurrent execution. To communicate between objects, to trigger execution of scripts, or scripts beyond objects, broadcast messages are used. By means of this mechanism, sequential or parallel execution of scripts is possible, either within the same object or over object boundaries. In addition to the basic control structures, Pocket Code offers event triggering building blocks for event-driven programming. Familiar concepts, such as variables, lists, or Boolean logic, are included. Furthermore, Pocket Code offers the ability to embed graphics, animations, music, and sounds.

In addition to Scratch, Pocket Code has a 2D physics engine integrated which enables the user to define certain physical features of objects and the stage (e.g., a simple collision detection, the definition of velocity, gravity, mass, bounce factor, or friction) to create from simple up to complex simulations of real world experiments. With Pocket Code's intuitive merge functionality, the new parts of two programs can be seamlessly merged together into one larger program — parts of the two initial programs that exist in both are not duplicated. This makes programming cooperatively, e.g., on different levels, much easier.

Modern smartphones are equipped with a large number of sensors, although most mobile games only use few or none of them (Kafai and Vasudevan, 2015). Within Pocket Code, users can create games using the device's sensors, such as inclination, acceleration, loudness, face detection, GPS location, or the compass direction, which makes user input easy and engaging. With Pocket Code it is also possible to connect via Bluetooth to Lego Mindstorms® robots or Arduino™ boards. The following extensions are available: Lego Mindstorms NXT/EV3, Parrot AR Drone 2.0 and Parrot Jumping Sumo Drone, Arduino, Raspberry Pi (via WiFi), NFC tags, Phiro robots[58], and Chromecast.

One new feature (at the time of writing still in development) is an extension to program embroidery machines. Once available, self-created patterns and designs can be stitched on t-shirts, pants, or even bags or shoes. With Pocket Code, the embroidery machines will be programmable, similar to the existing TurtleStitch[59] project, which realizes this concept on a PC (while with Pocket Code only a smartphone is needed). As a result, teenagers have something they can be proud of, something they can wear, and they can show to others. This feature has proven to be especially engaging for female teenagers and show them new ways of expressing themselves creatively through coding. Figure 3.2 shows an example of an embroidery pattern made with Pocket Code.

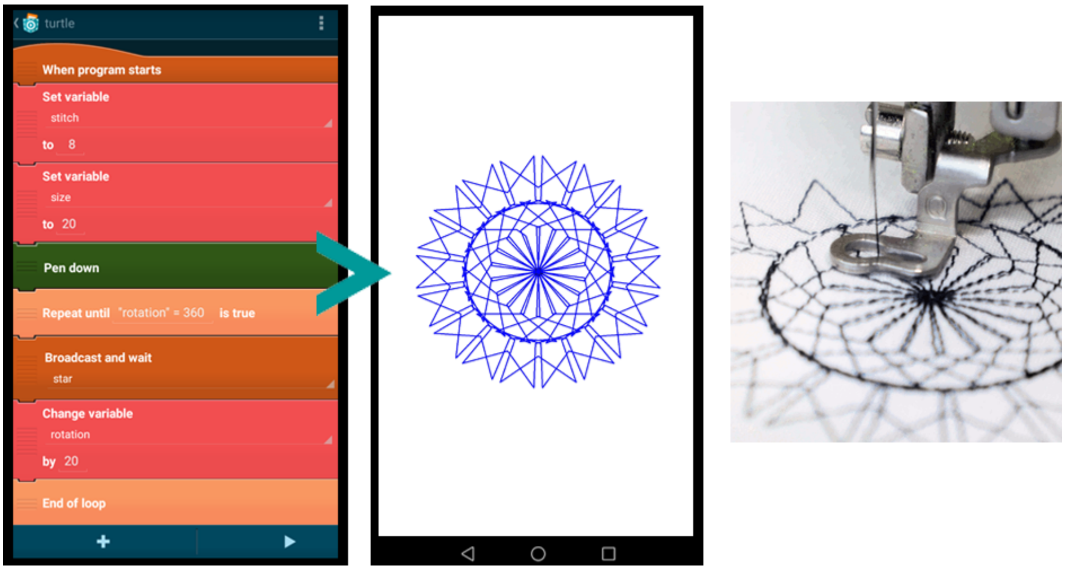


Figure 3.2.: Stitched patterns in Pocket Code. Picture in the right with kind of permissions from Andrea Mayr-Stadler, www.TurtleStitch.org project.

These kinds of computational construction kits make creating programmable hardware accessible to even novice designers and combines coding and crafting with a rich context for engaging students (Kafai et al., 2014). In the context of robots, being able to program a smartphone makes much more sense, as the smartphone can be mounted on the robots, thus allowing to give it a face, a voice and other sounds, and additional sensors can be used such as acceleration, inclination, magnetic field, GPS, voice recognition, and computer vision. Also, since only a smartphone is needed with Catrobat, the programming can be done on the spot, outside, e.g., when using one's land-based robot or flying drone outdoors. With Catrobat no laptop or PC is necessary, thus, coding can take place anytime and anywhere, and in particular can be widely made available even in less affluent communities around the

[58] Robotix: http://www.robotixedu.com/phiro.aspx
[59] TurtleStitch: http://www.turtlestitch.org/

world. In addition, Catrobat released a Scratch Converter to allow the conversion of existing Scratch projects to the Catrobat language directly within the app, so there are, currently around 30 million Scratch projects available for remixing and providing inspiration for our users.

Pocket Code interface consists of different areas: First, the app itself with a main menu and the program collection of downloaded or self-developed programs, and second a community sharing platform, which is integrated into the app as a web-view, and which serves as a learning, sharing, remixing, cooperation, and publishing place. Third the "stage" where programs get executed on the phone, and additionally a sophisticated graphical editing program that allows to draw and edit the looks of all actors, objects, and backgrounds of one's programs.

This **community sharing platform**[60] provides an online platform for users to download and upload programs, share them with other users, search for programs, and to provide feedback, e.g., write a comment to a program or rate a program. In addition, tutorials and starter programs are provided. The web application is based on the Symfony Web Framework, implemented in PHP, a web programming language, the data are stored in a relational MySQL database and it uses Doctrine as an object-relational mapping tool. In community website's program overview, users can execute the program directly in desktop browsers (HTML5 web player), download the program to the Pocket Code app, or download the program as a standalone Android app. In addition, the tool automatically creates statistics from Pocket Code programs and provides an online code overview. The program details page is illustrated in Figure 3.3.

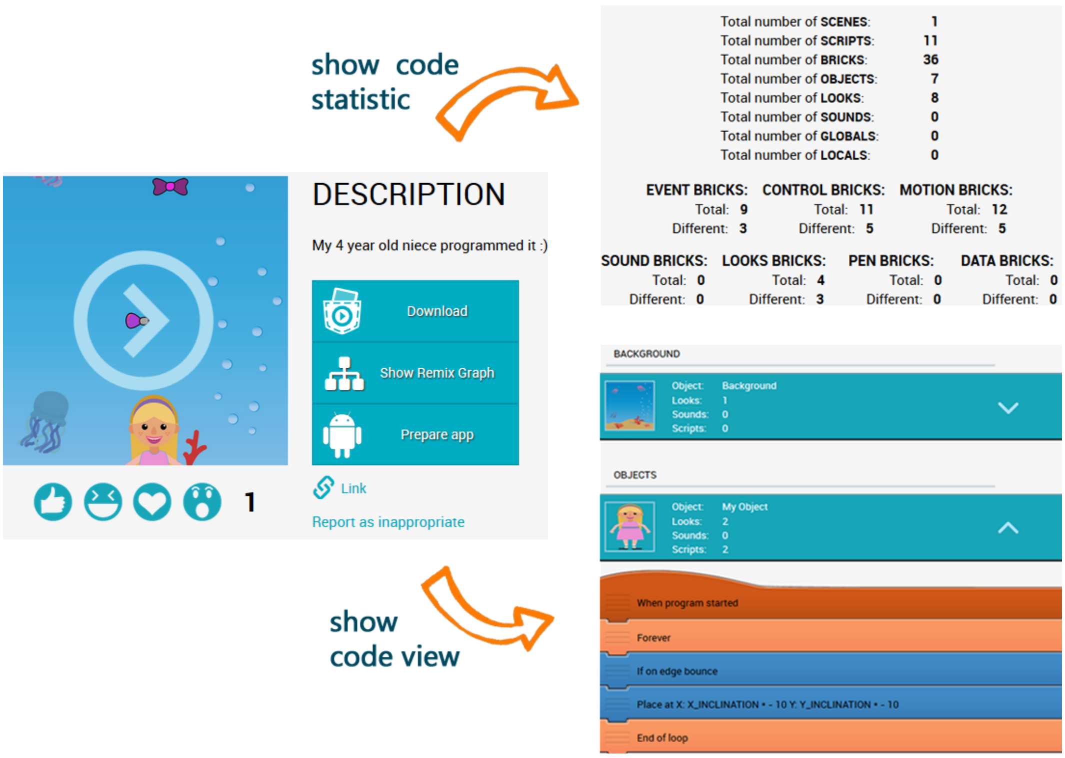


Figure 3.3.: Pocket Code web-share: programs details page with code statistic and code view.

[60] Community sharing platform: http://share.catrob.at

Figure 3.4 illustrates the options within the **main menu**.

Figure 3.4.: Main menu: 1. within the settings the user finds, e.g., the Accessibility Preferences or the Scratch Converter; 2. the user can create a new program by starting with an example game or with an empty game; 3. program overview: the user can open a program to execute or modify it; 4. open help: videos, tutorials, step-by-step tutorials, education page for teachers and students or Google groups forum; 5. download and play games from other users; 6. upload a game to the sharing platform.

If the user starts first with a new and empty program, it initially only contains one empty background object. With the “+” sign users are able to add objects, looks, or sounds (depending on which activity he or she is in). The background object itself can be assigned to several backgrounds, which can be exchanged during runtime. The background can also have its own scripts. Every program can consist of multiple objects and at least one background (which is a special kind of object). Every object can hold a. scripts that define the behavior of the object, b. looks which can be changed and used, e.g., for object animation, and c. sounds to make the object play music, other sounds, or recorded speech. Scripts can control the looks and sounds. Figure 3.5 holds an overview of a new Pocket Code program UI.

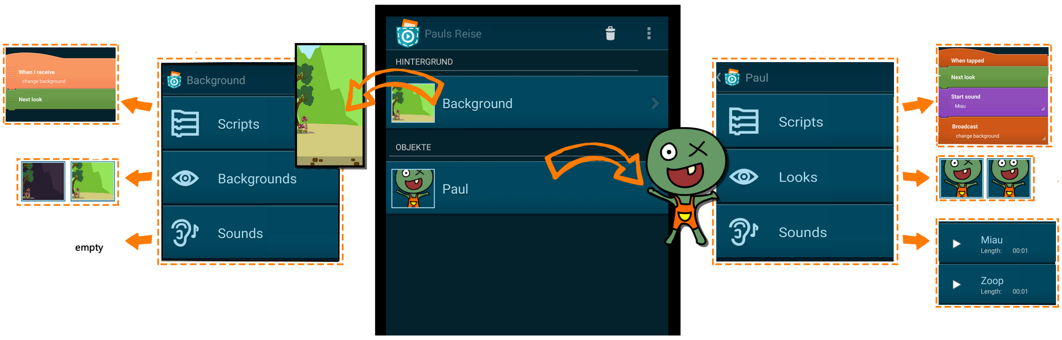


Figure 3.5.: Pocket Code's UI in general.

Looks can be drawn and edited with Pocket Paint. Pocket Paint[61] is a second app of Catrobat available on Google Play which allows users to create their own objects with a pencil or different shapes (note that the project currently work on integrating this second app completely in Pocket Code in order to simplify the installation for our users). Distinctive features of Pocket Paint include the ability to use transparency, to zoom in up to pixel level, to change the dimensions of the looks, and to use layers, the latter being particularly interesting to create consecutive looks from an animation series. In addition, users can add looks with their camera, from their phone's memory, or use the Catrobat's Media Library with a collection of predefined graphics. To add a new sound the user can either record a sound directly in Pocket Code, add a sound from the Catrobat Media Library, or add a sound from the phone's memory. This workflow is illustrated in Figure 3.6.

[61] Pocket Paint: http://catrob.at/PPoGP

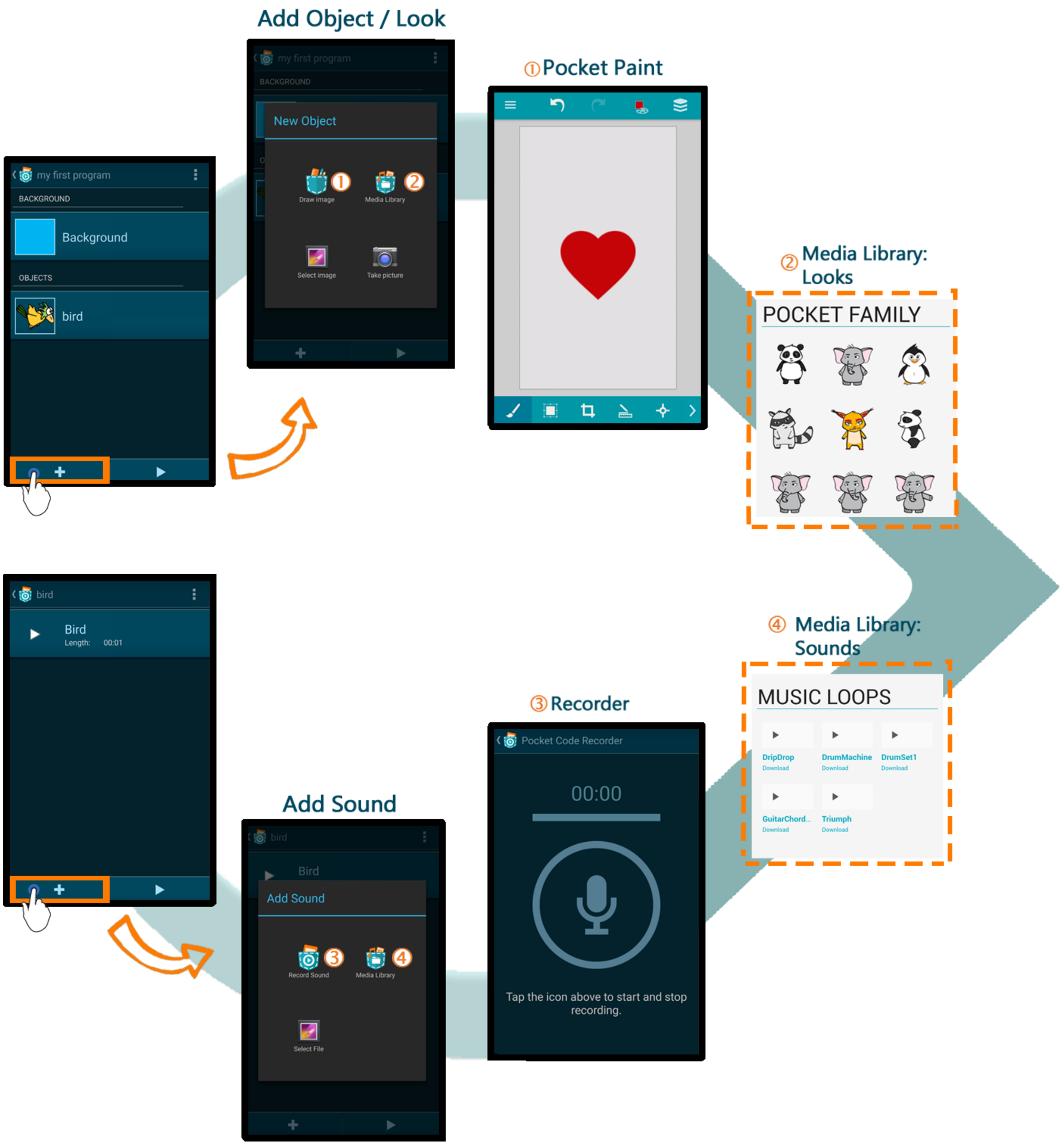


Figure 3.6.: Pocket Code's UI: add a lookobject or add a sound with the "+" sign.

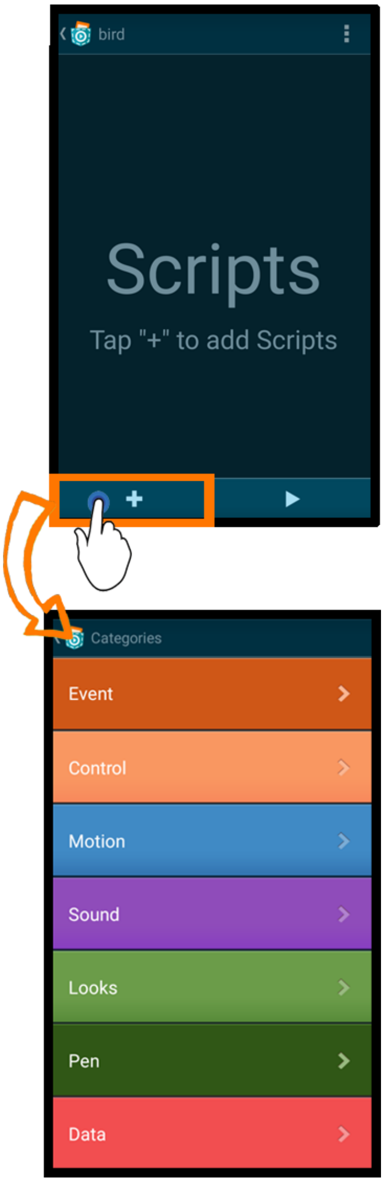


Figure 3.7.: Pocket Code's UI: add new scripts and choose bricks from the seven basic available categories.

A **script** is a collection of code blocks that contain the logic of programming and define the operations of the object. Thus, it is possible to move the object and access its properties and change them. For adding scripts there are seven different brick categories (see Figure 3.7) a. The Event category in dark orange that contains "hat"-bricks or broadcast bricks. Hat bricks are special kinds of bricks that, depending on certain circumstances such as a tap on an object, start the attached script; b. The Control category in orange contains if-then-else bricks, loop-bricks to control the flow of the script, bricks to switch between scenes, clone bricks, etc; c. The Motion category in blue color contains bricks to manipulate the object's position, orientation, or movements; d. The Sound category in purple contains bricks to start and sounds, manipulate the volume, or accept spoken input; e. The Looks category in green contains bricks to change the graphical appearance of the object, e.g., set/change size, brightness, transparency or hide/show the object as well as set a certain look to animate the object, or to show speech and think bubbles, or to ask for written user input; f. The category Pen in dark green holds bricks for drawing lines (a pen that follows the object) and the option to leave stamped marks of the object on the background; g. The Data category in red contains bricks to manipulate variables and lists, e.g., to set/change variables, maintain lists, add/insert/replace items, and show variable content on the stage. This color scheme makes is possible to understand scripts more easily through the bricks' color which supports readability. By activating extensions in the settings menu, additional categories appear for Lego (yellow), Drone (brown), Arduino, the Phiro robot (both in cyan), etc.

The **formula editor** looks like a similar to a calculator and allows the creation and execution of mathematical and logical formulas that can be used in bricks. The formula editor is shown in Figure 3.8. It consists out of an input field that show and compose the formula, a keyboard, and a compute button to display the current result. On the keyboard, five categories for various values, functions, and operators are available. a. Object: a collection of values of the current object, e.g., values for the X and Y coordinate, or the current speed, b. Functions, such as sin or cos, and a random number generator, or list and string functions, c. Logic is used to compare values or to combine logical expressions. d. in Device there is information that the smartphone or tablet records, e.g., inclination, loudness, or GPS data, and e. Data stores variables and lists and shows their last value.

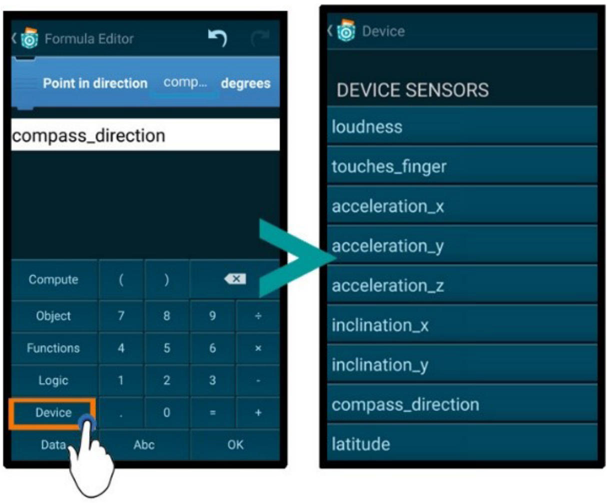


Figure 3.8.: Formula Editor: a. The value for the direction can be defined as a constant or, a sensor can be chosen by tapping "Device"; b. Available sensors which can be used.

With a tap on the play button the program starts. The objects are shown on the stage and the scripts are executed. To stop or to pause the program, the user has to tap on the back button of the phone. A stage menu appears which can be seen in Figure 3.9. The stage is organized in a logical coordinate system with an X- and Y-axes, which allows an exact positioning of the objects. These axes can be displayed in the stage menu (see Figure 3.8c.).

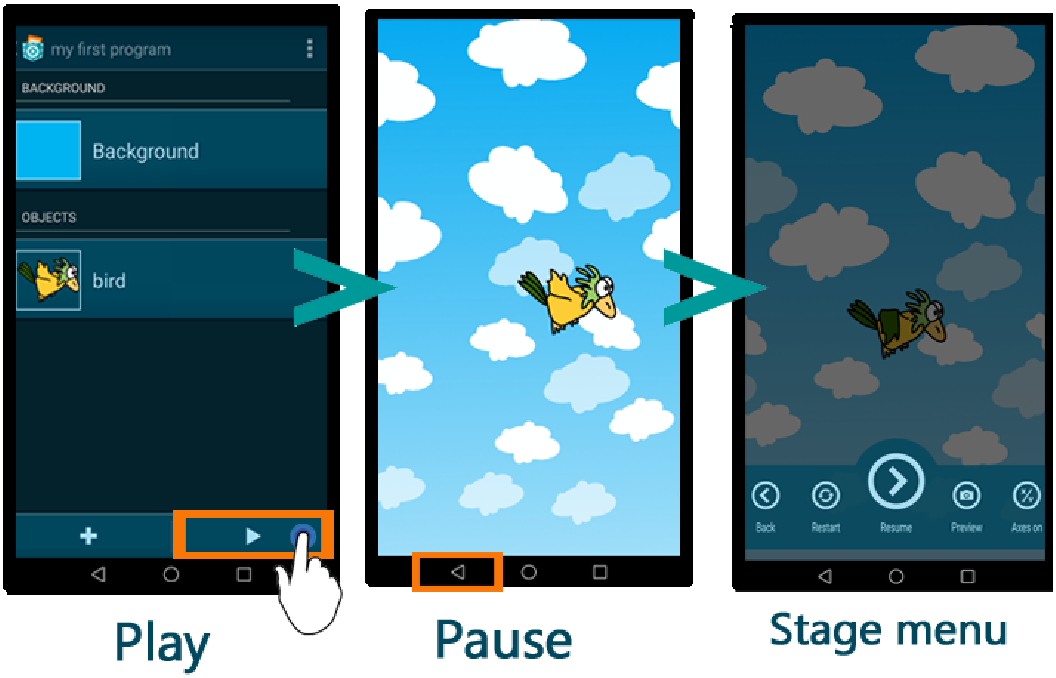


Figure 3.9.: Stage; a.) tap the play button to start the program, b.) tap the back button of the phone to pause the game, c.) in the stage menu the user has five options: 1. tap back again to stop the game and switch back to editing of programs, 2. restart the game, 3. resume the game, 4. add a new preview picture to the program (this will be shown, e.g., on the sharing platform), and 5. display the X/Y axes on the device screen.

3.2.2. Pocket Code example program.

To demonstrate the user interface and how the integration of sensor data can be achieved, a small Pocket Code program is presented in this section (see Figure 3.10). This example was part of the author's paper (Spieler et al., 2016). The goal is to create a bird flapping its wings that always points to North despite the phone's direction.

This demo consists of two elements: a sky-blue background object which is inanimate and a bird object (see Figure 3.10a). The bird object may contain scripts, looks, and sounds (see Figure 3.10b). In this demo it has two different looks which are used for the animation of its wings (see Figure 3.10c). The bird's scripts section contains a single script that makes the bird flap its wings and updates the bird's direction to the North (see Figure 3.10d).

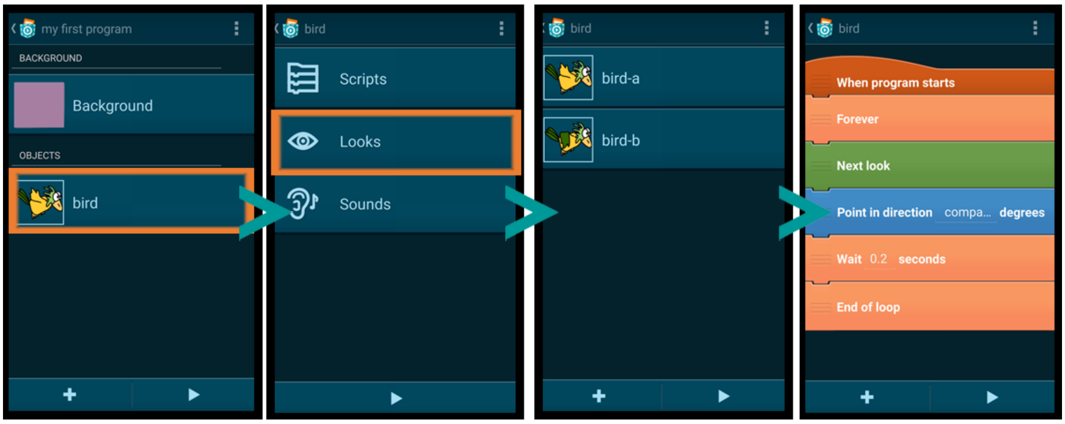


Figure 3.10.: Example program a. the program consists of one object and the background; b. the bird can hold scripts, looks and sounds; c. the bird has two looks used for animation; d. the bird's script defines its behavior.

The script in this demonstration (see Figure 3.10d) consists of five bricks from three different categories. The first brick "When program started" is a trigger, which starts the execution of the script whenever the Pocket Code program is started by the user. The "Forever" brick with its delimiter "End of loop" represents an endless loop, meaning that every brick between "Forever" and "End of loop" is executed as long as the Pocket Code program is executed. Within this loop the core components of the bird's behavior are defined. The "Next look" switches the object's appearance from "wings up" to "wings down" which gives an illusion of animation. The "Point in direction" brick updates the object's direction. To use the compass direction with the "Point in direction" brick, the formula editor must be invoked by tapping the brick and choosing "Edit formula" from the context menu or by tapping the direction value field. The value then can be edited with the formula editor (see Figure 3.9a). Tapping on to the "Sensors" field of the editor opens a list of available sensors (see Figure 3.9b). which can be inserted as a direction data provider for the "Point in direction" brick. The last brick's purpose in the forever-loop is to slow down the animation rate. Hence, a "Wait 0.2 seconds" brick is inserted causing a 0.2 second delay in this loop.

3.2.3. Pocket Code for academic purpose.

Pocket Code is an easy way to start coding or to be used in project work at schools. In addition, Pocket Code greatly facilitates the research since the relevant data (the programs) were recorded when uploading the games to the sharing community website.The advantages of using mobile phones in schools has been already described in Section 2.1.4.2. Pocket Code is not intended to develop standard applications, but to promote understanding the logic behind coding and foster conceptual thinking (see Section 2.1.4), thus following a constructionist approach in learning by doing and the creation of sharable artefacts (see Section 2.1.3.1). Thus, creative and artistic talents can be recognized and the learning occurs in a student-centered, project-based setting with the use of new media. An imminent feedback, by executing the code and seeing the result on-stage, makes detecting bugs in the program easier. Complexity can vary from simple sequences of steps to create an animated story to the use of traditional programming concepts (branching, loops etc.) to create interactive games. Pocket Code facilitates students in creating their own games within an educational context and can improve student motivation and engagement. Use of gaming technologies e.g., reward system, or other MDA elements (see Section 2.2.1.1), as well as automated analysis of user input/submitted project work, can help to motivate them further and shows interesting insights for teachers to provide effective feedback. Users of Pocket Code are mostly teenagers who can learn from each other and share their ideas to create new games and other apps together. The community sharing platform allows create a knowledge-building community and to give and receive feedback, support, and assistance from others around the world, thus allowing our users to stand on the shoulders of their peers and learn from each other.

Pocket Code aims to reduce the complexity of coding by giving students responsibility for their own learning process. Visual programming languages in general help novice users to express their ideas easily and create small programs within minutes (see previous sections). These languages typically keep the focus on the semantics of programming and eliminate the need to deal with syntactical problems. Through coding with visual programming languages like Catrobat, an insight can be gained into how students think and their cognitive processes can be visualized (Papert, 1985). As described in Section 2.1.4, coding supports logical thinking skills and this is important to students in many ways, e.g., when making important decisions (Syamsul and Norshuhada, 2010). The syntax of traditional programming environments is usually too complex for children to understand or it does not fit their interests and experience.

> “In particular, we felt that it was important to make the floor even lower and the walls even wider — but still supporting the development of computational thinking.” (Resnick et al., 2009, p. 62)

Thus, visual programming environments provide an easy start, but allowing to develop more complex projects as well by adding “Wide Walls”. This emphasizes that Scratch supports a wide variety of projects as well as ways to learn and play, according to the needs and interests of its users (Resnick, 2017). Teenagers can try out new ideas and realize the projects they define for themselves, aided and inspired by likeminded others in a user-friendly and social environment. Catrobat fosters diversity and learning in a worldwide community. It is important that teenagers can personally contribute, e.g., with their own photos, graphics, music, or recordings. As the theory of constructivism states, it must indicate something meaningful for them and bring interesting insights. The social part is covered by giving the opportunity to share the programs on the community sharing platform (see Figure 3.4). Students love to publish their own work, e.g., on YouTube (Wolber, 2009). According to the constructionism theory, an essential part of learning is to present something self-constructed to the public.

To conclude, it suggests that teenagers may see Pocket Code as a welcome addition to traditional classrooms and that parents may see pedagogical value in time their children spending on their phones engaged in creative fun activities.

3.2.3.1. Pocket Code to reinforce female teenagers.

More and more female teenagers play mobile apps (see Section 2.4.2.4) and the number correlates with the popularity of smartphones among this group (see Figure 3.11). Thus, Pocket Code is an interesting tool for them to be creative, to collaborate with other as well as to learn. Thereby, Pocket Code fosters intrinsic motivators for inclusive environments (see Section 2.4.2), like providing a sense of belonging with an existing community that shares, remixes, and creates games with a wide range of different topics. The focus on game design, creativity, sharing one’s assets, and using sophisticated assets provided through the Pocket Code's Media Library make coding more interesting and fun. With only a few bricks, animations, narrative stories, interactivities, or movements can be easily included in games and provide self-efficacy by reaching small steps in creating a bigger game during project work. With Pocket Code female teenagers can be reinforced, e.g., by guiding them and supporting them by asking them specific questions to foster their CT skills, playing an active part in problem-solving strategies, and finally by providing recognition of their work done.

Teamwork and a sense of ownership are two important factors of motivation but not mutually exclusive to each other. Working on their own levels and later adding these levels to a bigger game fosters inclusion, collaboration, and engagement for girls. To foster female students in their coding as well, tutorials and step-to-step guidance needed to be developed to help them with important steps. To conclude, at the beginning of the project, Pocket Code already had huge potential to reinforce female teenagers through playful activities but needed more focus to truly understand needs and expectations of this specific target group of young girls.

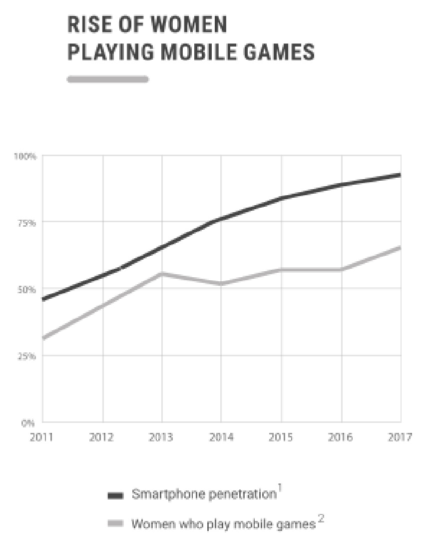


Figure 3.11.: Rise of women playing mobile games (Google Inc., 2017).

3.2.3.2. Pocket Code an optimal tool to perform game jams.

In Section 2.2.2.3 the author presented the advantages of running game jam events in schools. Since no programming experience is needed to use the Pocket Code app, it seems to be a very suitable tool for novices who attend a game jam as it allows for creating playable games within a short time span in fast-paced and collaborative environments. There are several key advantages to using Pocket Code for game jams. For example, the feature to merge programs among users and transfer objects, code, looks, and sounds between projects via the "backpack" functionality fosters distributed development and collaboration among students. Using the constructionist approach (Papert and Harel, 1991) and combining it with the game jam idea (Kaitila, 2012) is new opportunity to enhance collaboration and teamwork. At the beginning of the project, the team conducted and attended game jams to gather insights in how to run such events. Thus, the team identified three challenges connecting game jams with Pocket Code. First, there were difficulties working in teams on one program, and second, finding a suitable time frame for game jams. The GGJ event has a time frame of 48 hours but in a school context, it is easier to conduct game jams over a longer but less intensive period of time, where students are able to work on games, e.g., as a homework exercise over several weeks. The third issue was finding appropriate sizes of participating teams.

Statistics presented in Section 2.1.4.1 show the wide distribution of smartphones among our target group and supports the argument of using smartphones for game jams for this target group. Thus no costly hardware or programs are needed for participation. This point facilitates the setup of Pocket Code game jams since there is only a minimal organization effort for teachers and schools. Furthermore, global participation is possible. Since Pocket Code is translated into more than 40+ different languages, students all over the world can participate and use the app in their mother tongue. Students worldwide can be reached through online game jams and submission through the website. Moreover, participants can be more or less experienced. Thus, it was essential for the jam to provide themed tutorials, guidelines and an easy workflow to submit games. The setting of our performed game jams during NOLB will be presented in Section 3.4.1.1 and the evaluation in Section 4.5.

Our new and forward-thinking approach to code on mobile devices received national and international recognition. The Catrobat project has won a number of awards, including 2016 two Lovie Awards ex aequo with Red Bull and Doctors without Borders, evaluating the best European digital projects in London, and the Reimagine Education Award for innovative educational projects at the Wharton Business School of Pennsylvania[62]. Additionally, in March 2017, Catrobat won the "Platinum Award" in Best Mobile App Awards Best Educational App category and 2016 the “Internet for Refugees” award for a Right-to-Left language implementation of Pocket Code, which supports RTL languages, e.g., Arabic or Farsi, and particularly focuses on refugees and children in crisis or development areas. A new project “RemoteMentor” was started January 2018 which promotes remote mentoring by connecting female role models with female programming beginners. This idea was awarded with the "Closing the Gender Gap"[63] prize of the Austrian NetIdee in November 2017. In addition, Google is supported the project by featuring it on Google Play for Education and it has already been accepted six times for the Google Summer of Code (GSoC) program[64] and will return again in 2018.

3.3. European H2020 Project: No One Left Behind (NOLB)

The European No One Left Behind[65] (NOLB) project has been funded by the Horizon 2020 framework and involved partners from Germany, Spain, the UK, and Austria. Figure 3.12 illustrates the key elements of the NOLB project.

[62] Reimagine Educaition Award: http://www.reimagine-education.com/awards/reimagine-education-2016-honours-list/
[63] NetIdee: https://www.netidee.at/node/625
[64] Google Summer of Code: http://developers.google.com/open-source/gsoc
[65] No One Left Behind project: http://no1leftbehind.eu/

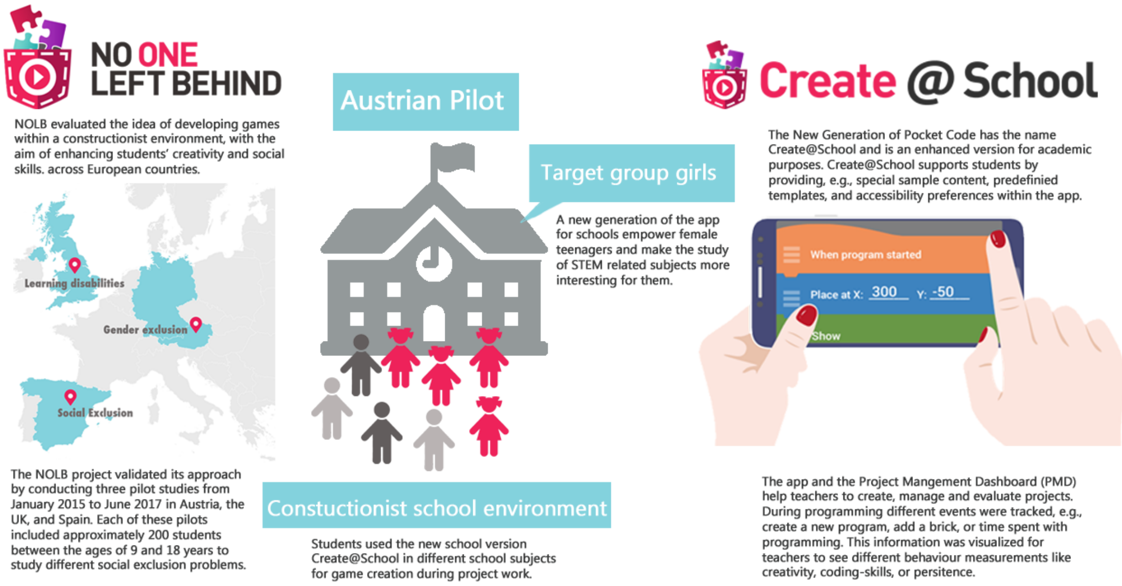


Figure 3.12.: The NOLB project, the Austrian pilot and the Create@School app.

The vision of the NOLB project was to unlock inclusive gaming creation and to construct experiences in formal and informal learning situations from primary to secondary level, particularly for children at risk of social exclusion. The focus laid on those teenagers with special educational needs and disabilities (in UK) as well as immigrants (in Spain); the project was also used as a chance to recognize gender differences in engaging with coding activities and game jams (in Austria). The project aimed to develop a new generation of Pocket Code, which should be a more mobile media-rich programming environment for teenagers, by providing meaningful learning experiences and supporting learners to realize their full potential. This was done by transferring game mechanics, dynamics, assets, and in-game analytics from non-leisure digital games into the app and by evaluating the coding lesson. This project started in January 2015 and reached its conclusion in June 2017.

The project goal was threefold: First, the team studied new ways of learning by focusing on a constructionist approach (see Section 2.1.3.1). Students solved curriculum-related problems while using the Pocket Code app, e.g., in physics, computing, arts, music, history, or language-learning courses. This interdisciplinary approach allows teachers to integrate coding not only in computer science but also in interdisciplinary and regular subjects which is a commonly used concept in European schools (see Section 2.1.1). Furthermore, for NOLB, teachers guided and assisted students in their learning processes by using a GDBL approach through making games to topics from their subject areas (see Section 2.2.2.2). Students were supported to take control of their own education, and became more engaged, interested, and empowered as a result.

Second, the NOLB partners created an enhanced version of the app Pocket Code and adapted it for use in schools. This new version developed during the project was called Create@School and integrated the results of the observations during the pilot studies as well as considered feedback from both teachers and students. Furthermore, a new web-based Project Management Dashboard (PMD) for teachers was developed. The app was released in October 2016 as an open Beta version on Google Play for first test runs during the Second Cycle of the project.

Third, the team investigated Pocket Code's potential for engaging students at risk of social exclusion. Thus, three small scale experimental pilot studies in Spain, the UK, and Austria were performed. The target groups and pilot schools are described in more detail in the next sections.

This game creation challenges in classes should enhance students' abilities across all academic subjects, including logical reasoning, creativity, and the development of social and computational thinking skills (see Section 2.1.4.1). Thus, the team studied whether the use of mobile game design has an impact on learning, understanding, and retention of knowledge, as well as satisfaction for students at risk of social exclusion. Moreover, students had the opportunity to socialize with their peers during the game-making process by working in teams. One outcome in Austria included more than 400 sharable programs designed to reflected students' different styles of thinking and learning (Ramnarine-Rieks, 2012). This section provides an overview about the setting within the different pilot countries, especially the setting in Austria, and describes the experimental cycles of NOLB.

3.3.1. A cross-european case study.

The outcomes during NOLB have been validated in three European small-scale experimental pilots in Spain, the UK, and in Austria on primary and secondary levels. The project was performed in three phases and each site targeted approximately 200 students between 9 and 18 years old experiencing different social exclusion problems. Hence, the Pocket Code app has been integrated into different curricula subjects and improved for academic use in every phase of the project.

Partners within each country focused on a different type of learner:

- *Austria* focused on enhancing girls' interest in STEM subjects through fostering social inclusion in class communities (see results in Delivery 5.4 (Spieler and Mashkina, 2017))
- *Spain* targeted on learners who are at risk of social exclusion related to immigration through fostering collaboration, cooperation, and engagement between Spanish and immigrant students (see results in Delivery 5.5 (Gaeta and Cea, 2017))
- in the *UK*, the team was working with children who have special educational needs and disabilities (see results in Delivery 5.3 (Hughes-Roberts et al., 2017))

To limit the scope of topics to those relevant to this thesis, the remainder of this section only focuses on the Austrian pilot. The description and results of the pilots from UK and Spain would be outside the scope of this thesis and they are not discussed in more detail.

3.3.2. The setting of the Austrian pilot.

This section provides an overview about the participating pilot schools in Austria, students that took part in the study, and teachers and their background and subjects in which Pocket Code was used are described in more detail.

3.3.2.1. Pilot schools.

In Austria, the NOLB project was piloted in three different schools situated in and around Graz. All three participating schools were AHS — Academic Secondary Schools (for more information about the different school types in Austria, see (Federal Ministry of Education Austria [4], 2017)).

The first school was "*Graz International Bilingual School*" (GIBS)[66], a bilingual (German-English) senior secondary school for students aged 10-18. Teachers at GIBS help the students to apply English with an intensive phase of language acquisition in the first grade. This international character should help them (as stated on their website) to "*encourage respect for cultural diversity and in developing critical awareness and constructive ways of dealing with criticism of oneself and the world at large.*" GIBS participated with eleven different classes, which ranged from Grade 6 to Grade 10 (students aged 12 to 16 years).

The second school was "*Akademisches Gymnasium Graz*"[67], which had already worked for several years with Catrobat. Computer science is taught from the first grade within the subject German and also in the fifth grade. The lesson plan included robotics, movie creation, and basics in coding (with C#, Pocket Code or web design). In this school, three different classes took part (8 courses) from Grade 6 to Grade 12 (students aged 11 to 18 years).

The third school was "*BORG Birkfeld*"[68] at Birkfeld, which is an one hour drive away from Graz. This school offers the possibility to focus on natural sciences and information technology from Grade 5 to 8. Students learn how to set up and manage computer networks, as well as basics of programming for computers, tablets, and smartphones. This school took part with three classes (Grade 9, students aged 14-15 years).

3.3.2.2. Participating students.

In total, 478 students participated in the Austrian pilot (281 female students, 197 male students; see Table 3.1).

Table 3.1.: Classification by gender and cycle. Feasibility Study (FS), First Cycle (FC), Second Cycle (SC)

Cycle	**Girls**	**Boys**	**Total**
FS, FC	165	107	**272**
SC	116	90	**206**
Total	**281**	**197**	478

The coding lessons were performed at the three schools described in the previous section with classes from Grade 6 to Grade 12 (11 to 17 years old). Figure 3.13 shows the amount of students per school and the distribution among genders.

[66] GIBS: http://www.gibs.at/
[67] Akademisches Gymnasium: https://www.akademisches-graz.at/
[68] Borg Birkfeld: http://www.borg-birkfeld.at/

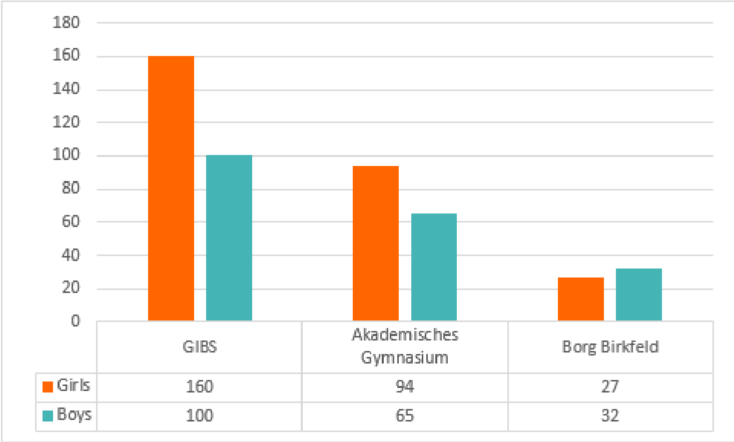


Figure 3.13.: Number of students per school and gender distribution.

Figure 3.14 shows the distribution by gender and grade level. Most students had Pocket Code lessons in Grade 9. The reason, as described in Section 2.1.2, is that computer science is only mandatory in Grade 9 for one hour a week. Since most of the participating teachers had their profession in informatics (see next section), it was easier to apply Pocket Code to this units than in units of lower grades.

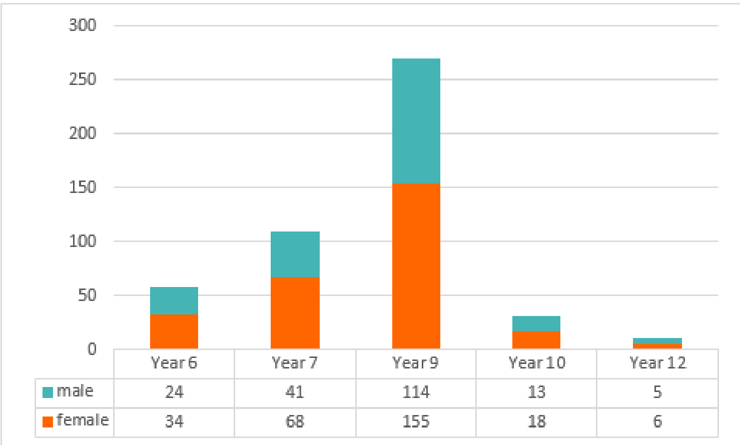


Figure 3.14.: Distribution of students by gender and grade level.

3.3.2.3. Participating teachers.

Eleven teachers participated on a voluntary basis in the Austrian study and were invited to use the app and materials during their classes. The description of the courses were part of the author's paper (Spieler et al., 2017). They provided feedback for improvement in a later stage. The teachers' gender, their technical background and a list of their subjects in which Pocket Code has been integrated, is visualized in Tables 3.2, 4.1, and 4.2.

Table 3.2.: Teachers of pilot school one/GIBS: AHS — A Bilingual school.

Teacher Nr.	Gender	Technical background	Subject in which Pocket Code has been integrated
1	female	no	English
2	female	no	English
3	male	yes	computer science
4	male	yes	computer science
5	female	no	physics
6	male	no	fine arts

Table 3.3.: Teachers of pilot school two/Akademisches Gymnasium: AHS.

Teacher Nr.	Gender	Technical background	Subject in which Pocket Code has been integrated
1	female	yes	physics, computer science
2	female	no	fine arts, computer science

Table 3.4.: Teachers of pilot school three/Borg Birkfeld: AHS with focus on computer science.

Teacher Nr.	Gender	Technical background	Subject in which Pocket Code has been integrated
1	male	yes	computer science
2	male	no	music
3	female	yes	computer science

To preserve the anonymity of the teachers, every teacher has been assigned to a number. This number is used in the description of the courses as well (see next sections) and are referenced in the evaluation chapter.

Before the start of the Feasibility Study (Fall 2015), an online questionnaire was set up to collect information about the teachers' digital skills and abilities (participating teachers: 1, 2, 3, 4, 7, 8, and 13). Therefore, the project adapted a Teacher Questionnaire, developed by Leicester City (Fraser et al., 2013), to focus on programming and gamification. This helped to gain better understanding of how resources such as training materials can be optimized to fit teachers' needs. The results of the questionnaire showed that all of them had already used digital media in their classes, e.g., presentations, links, and video clips, and all except two teachers (7, 8) also used boards/forums, blogs, or other tools for group discussion, or interactive elements, such as mobile quizzes or polls. Five out of seven mentioned that they used the school's Management Information System (MIS) to electronically record and monitor information about students' attendance, behavior, and achievements. Two teachers (1, 7) mentioned that they play computer games for fun and enjoyment and already used computer games while teaching their classes.

3.3.3. Experimental cycles.

The project validates its approach in different phases of the project: Preparation phase, Feasibility Study, First and Second Cycle. The main achievements of each phase are illustrated in Figure 3.15 and described in more detail below.

Figure 3.15.: Phases of NOLB.

3.3.3.1. Preparation phase (Jan.-Sep. 2015).

During the preparation phase, several meetings with teachers from different pilot schools were conducted. The goal was to gather information about the participating teachers and their subjects and to create suitable use cases for them to include Pocket Code into different school subjects. In all piloting countries, workshop days have been conducted at the end of the semester. In Austria, these workshops took place in June and July 2015 to kick of the Feasibility Study. Within these workshop days, Pocket Code was introduced to students, example programs were shown, and students got the chance to create their first games (these students did not attend during the NOLB cycles). In order to ensure a positive first experience with Pocket Code, the workshop was prepared with a lot of care. The theme for the workshop was “150 years of Alice in Wonderland” in reference to the 150th anniversary of Lewis Carroll's book, which was celebrated in 2015. This topic seemed interesting to the target audience: students 14-17 of all genders. For the workshop itself, various materials were prepared, including media assets, such as graphics, sounds, and tutorial cards for important functions (e.g., how to detect collisions). The "Shape of a Game" (see Section 2.2.1.2) framework was preinstalled on the mobile phones students used for the workshop, so that they could merge it into their games. With this framework, students' game design process was scaffolded, by allowing them to focus on the game development itself. In Austria, two pilot schools attended the workshops. At GIBS, 19 students took part and at Borg Birkfeld, 35 students attended the four hour workshop. The introduction session was held in front of the whole class to show what could be achieved within Pocket Code, with the team presenting example games and the user interface. Subsequently, students formed groups of two or three and pitched game ideas. These ideas were shared with the class and students got direct feedback. Next, students created storyboards, which helped them to get a clear image of the gameplay and characters. Without giving them further guidance in programming, they started to code their games. The team took the role of coaches, and students could consult them when they needed some input. This learning by doing approach supports the constructivist theory, and most importantly, students had the opportunity to add their creativity to this session and explore the various functionalities of Pocket Code on their own.

Figure 3.16 illustrates different impressions of the workshops. At the end, students uploaded their games to the community sharing platform and presented their games in front of their peers (with the use of a projector).

Figure 3.16.: Pictures of the workshop: storyboard, instruction session and framework "Shape of a Game".

The games created have been analyzed to define what kind of game design elements were used most often in order to generate tutorials cards, tutorial videos and frameworks with Pocket Code. In both workshops, many different games were created, e.g., mazes, skill games, jump and run games, quizzes, adventure games, or racing games. 98% of the students described the workshop as very good or good, 90% were satisfied with their results, and 89% liked to work in teams. These workshops were the first steps in regard to the Feasibility Study and these first experiences were very important for the later development of the lesson plans and the structure of the Pocket Code lessons. This will be part of Section 4.3.1.

Before finalizing the planning phase, the teachers needed to be involved as well. The success of the project depended a lot on their participation and cooperation. Influencing factors included the subjects in which teachers were planning to use Pocket Code and the amount of units available. In September 2015, the first teacher trainings were held to show the functionalities of Pocket Code. In the two-hour courses, teachers were given a short introduction to Pocket Code, after which they created their first program with the workshop facilitator. The hands-on session was done in a step-by-step approach based on the teacher course that was designed by TU Graz[69]. The sessions were followed by short discussion rounds where teachers considered how Pocket Code could be used best to support their lessons. All training sessions were successful, and afterwards the teachers created some excellent ideas together with their students. In the following weeks, they sent these ideas either in the form of storyboards or descriptions to the NOLB team. After both the teacher trainings and the student workshops, the following materials were prepared to kick off the Feasibility Study:

- Online course including explanation of the bricks, tutorial cards for students, and helpful game metrics (“Shape of a Game”, game design process)[70]
- Short video tutorials with gaming concepts[71]
- Pocket Code frameworks (programs) for physics, fine art, English, computer science, and music[72]
- Media assets like graphics and sounds online available through the Catrobat Media Library[73]

3.3.3.2. Feasibility Study (FS) (Sep.-Dec. 2015).

In September 2015, the first pilots for the Feasibility Study started. The target was to test how Pocket Code can be used in a wide range of curricula subject, e.g. mathematics, geography, religious education, fine arts, physics, computer science, or music. In particular, the feasibility pilot helped to identify students’ and teachers’ specific needs for those that have special educational needs and disabilities, are disengaged learners, and/or at risk of exclusion related to the project’s objectives. All this influenced the full pilots and the planned Pocket Code enhancements, which shaped the new generation of the app. Figure 3.17 shows the different stages of the Feasibility Study (Tinney et al., 2015).

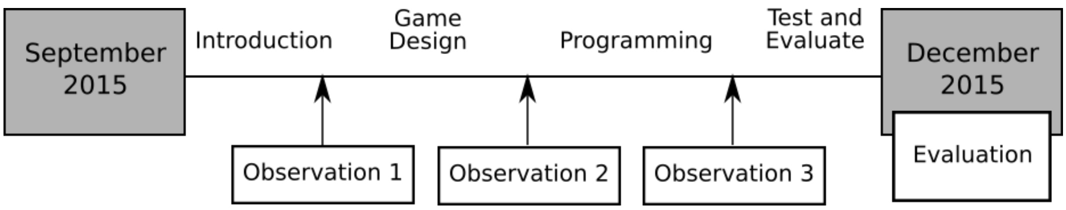


Figure 3.17.: Schedule of the Feasibility Study (Tinney et al., 2015).

[69]Education platform: https://edu.catrob.at/fundamental-online-workshop
[70]Tips: https://edu.catrob.at/documentation-tips
[71]Tutorials: https://share.catrob.at/pocketcode/gaming-tutorials
[72]Tutorials Feasibility Study: https://share.catrob.at/pocketcode/profile/3611
[73]Catrobat Library: https://share.catrob.at/pocketcode/media-library/looks

During the "Introduction" the team established exact curriculum elements and learning objectives, like described further in Section 3.5 "Observation 1" (which here means on-site observations) included issues, such as curriculum, pedagogical, teaching, technical, and any other problems or adaptations necessary. In this phase a survey was conducted (see Section 4.2). "Observation 2" followed up with an analysis of the students' ideas. In "Observation 3", the team measured Pocket Code's usability and the intention to use through a second survey. In addition, the engagement and attainment level was observed (during on-site observations and teacher notes). During the "Test and Evaluate" phase, the completed games were tested and evaluated and the outcomes assessed against learning objectives (see Section 4.4.3). The "Evaluation" phase showed if the process met the learning objectives as set by the teachers. In addition, one month after using Pocket Code during school a final survey was conducted. Moreover, during the Feasibility Study, the Alice Game Jam event was performed. Results of this event are part of Section 4.3.2.1. Furthermore, frameworks, tutorials, and resources were reviewed and adopted.

When integrating a gaming approach into school contexts, there are a few challenges to consider (Jaime and Ruby, 2011). The following challenges were identified when introducing the game-based approach into the school context during the Feasibility Study, (see Section 2.1.4.2).

Stick to the curriculum: One important task was to identify how a particular game can fit into components of the statutory curriculum as well as the accuracy and appropriateness of the content within the game. The difficulty was in assessing any learning gains delivered as a result of the game-making process.

Time and curriculum constraints: Another key factor was the limited time available for teachers and students to familiarize themselves with the game environment. To ease the time pressure during the school year, the team conducted several teacher meetings and trainings during the school year.

Technical constraints: The technical infrastructure of the school facilities and the availability of school hardware is essential. In order to ensure that all students have smartphones or tablets to work with, we have distributed a hardware questionnaire to gather information on what kind of smartphones are currently in use, and to discover how many devices are needed. All partner schools got 30 tablets per school for the project period.

Assessment of the outcome: It was important to ensure that the technology used actually improves teaching and learning outcomes and to provide students with appropriate feedback. The assessment is part of the evaluation chapter.

3.3.3.3. First Cycle (FC) (Jan.-Aug. 2016), and Second Cycle (SC) (Sep. 2016-Jun. 2017).

During the First Cycle, the NOLB partners evaluated preliminary results and continued working with Pocket Code. Furthermore, the team started preparing to adopt Pocket Code to the new version called Create@School and planned new infrastructural services, e.g., a teacher dashboard, in-game analytics, and the generation of statistic information. The key elements of the new version are presented in Section 3.4.2.

During the Second Cycle, the team conducted further coding activities with Create@School app, and used its new features. During this time, the app was repeatedly reviewed through interviews, surveys, data tracking, focus group discussions and programs submitted by students (see Section 4.3). In addition, in Austria, background material and information about female teenagers has been gathered, both from the literature as well as through interviews and focus group discussions with teenage girls.

In addition, at the end of 2016 the Galaxy Game Jam event took place. Results of this event are part of section 4.5.2.

3.4. NOLB: Developed Frameworks, Resources, and Tools

Within the NOLB project, it had been examined how to attract, motivate, and engage students with content from academic curricula (Smith et al., 2016). At the same time, it was ascertained how to support their learning processes to provide an effective learning experience first with Pocket Code and subsequently (during the Second Cycle) with the new tool, Create@School. Therefore, the NOLB team has developed different resources, frameworks, and tools, which interlink the use of Pocket Code within a games environment into the classroom through the development of games within different subject disciplines. Frameworks and resources compromised, e.g., the NOLB lesson plan, an interactive training guide, or the Game-Making Teaching Framework. The development of the pedagogical frameworks was mainly done by the UK-based team; however, they had to be applied in every pilot country. To explain the whole creation and evaluation of all of those tools would be beyond the scope of this thesis. Outcomes are part of the NOLB Delivery 5.3-5.5 (Hughes-Roberts et al., 2017; Spieler and Mashkina, 2017; Gaeta and Cea, 2017) submitted at the end of the project. However, a brief introduction of the NOLB concepts is given below. During the preparation phase of the project, one additional framework was added to the teacher's guide: The concept of game jams. Game jams were mainly developed and included in Austria. Thus, the main work was done by the TU Graz team and will be described in more detail within this chapter. The author of this thesis had the main lead on the development of Create@School app. The different parts of the app are described at the end of this section (see Section 3.4.2).

The NOLB *teachers' guide* has been structured in three steps: Plan, Teach, and Improve, inspired in the 'plan, do, check (or study), act' methodology (also known as Deming Circle (The Deming Institute, 2018)). Thus, the team ensured that games' mechanics and dynamics, and the academic content could be transferred in a 'fun' but pedagogic manner through a continuous improvement of the teaching processes. In addition, the NOLB team has used the Universal Design for Learning (UDL) principles to link it with the *what*, *how* and *why* of education, (see Section 2.1.3) and constructed its courses in reference to the constructionist approach. The PDCA and the UDL created the basis for the three pillars of the developed NOLB Game-Making Teaching Framework (GMTF), which is part of the next section.

3.4.1. Game-Making Teaching Framework (GMTF).

The NOLB GMTF framework provides a coherent approach to learning and teaching by integrating leisure-oriented gaming methods into multi-discipline curricula. One part of this framework is the integration of game-based methods via game templates that refer to didactical scenarios that include a refined set of genres, assets, rules, challenges, and strategies (see Section 3.4.2.2). This allows the transference of the components of games to the educational domain. The GMTF is part of the NOLB Delivery 3.2 (Martinovs et al., 2017) and part of the paper (Spieler et al., 2017).

The parts of the GMTF are a collection of day-to-day and core practices, gathered by the UK team from interviews and focus groups performed with teachers at the pilot schools. The framework is illustrated in Figure 3.18.

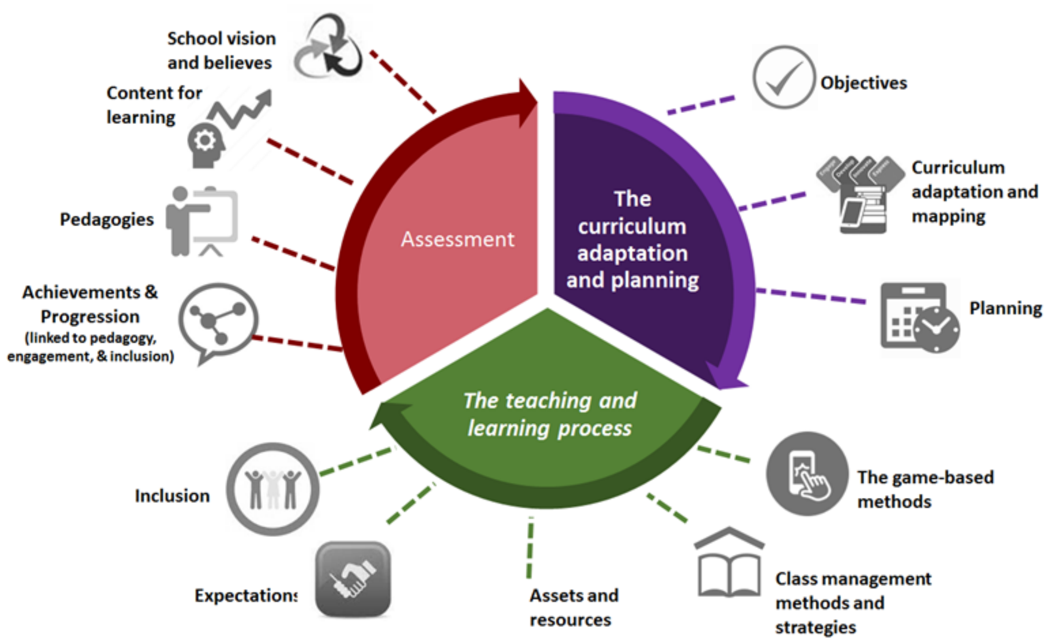


Figure 3.18.: Components of the GMTF in NOLB (Smith et al., 2016).

The curriculum adaptation and planning, as well as the visualization of the teaching and learning processes, helped to create the illustrative scenarios and to integrate game-based strategies. The assessment and feedback process is supported by the developed PMD, which includes the analytic dashboard. By merging *the what* and *the how* of learning, the result will be the following: a) clear objectives for each class and subject, and b) standardized game-based methods, including game mechanics and dynamics. For example, it was necessary that the developed game templates were applicable to several subjects and classroom levels (see Section 2.1.1), and universally recognized as "games" by students who are familiar with best practice game examples (see Section 2.2.1.1), in order to link game mechanics and game dynamics with academic content. As the games are taught through different subject areas, the learning objectives are also included in the lesson plans. An example of a lesson plan is presented in Appendix B. Beside the pre-defined templates another output of the GMTF was the Pocket Code training guide which is intended to show how to create a sequence of games and to show teachers the basic systems to navigate through the app. The guidelines explain some of the application's features, including several ways of creating an object or sound, making it interactive. The training guide includes a range of video clips, which is part of the education website[74].

For NOLB, the UDL principles were linked to the GMTF, so that it responds to the basic questions of learning: what is being taught, how the information is shared, and why the information is engaging the learners. Based on these three UDL pillars, the GMTF refers to: curriculum adaptation and planning process, the teaching-learning process, and the assessment and feedback process (see Figure 3.19).

[74]NOLB teacher space: https://edu.catrob.at/no1leftbehind-for-teachers

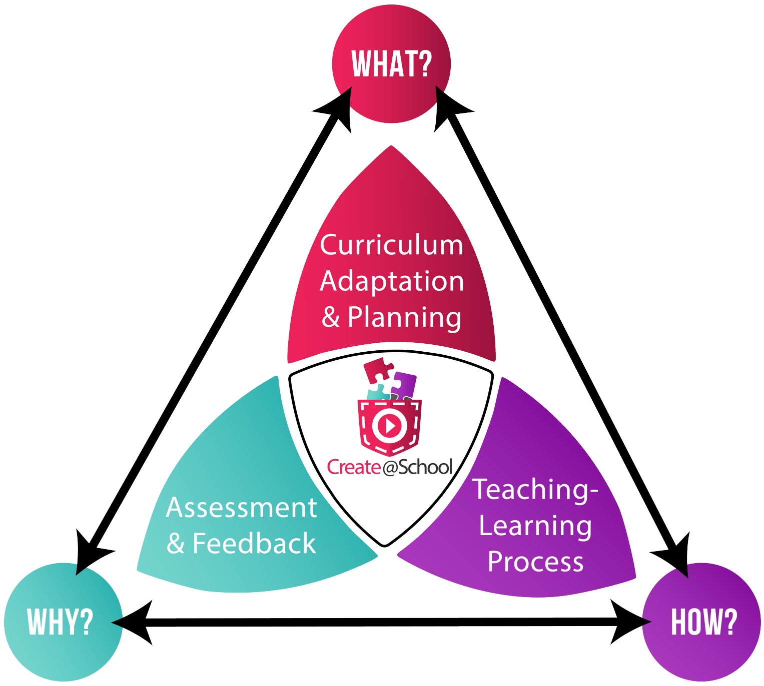


Figure 3.19.: Link UDL principles to the NOLB GMTF (Spieler et al., 2017).

In practical terms, this framework offers a multidisciplinary approach to:

- Creation and planning of game-based subject-relevant scenarios for templates (*what* they learn) e.g., physics, fine arts, music, etc.
- Review and integration of elements of game mechanics and dynamics into the teaching and learning process (*how* they learn)
- Assessment of relevant teaching and learning experiences (*why* they learn) and feedback

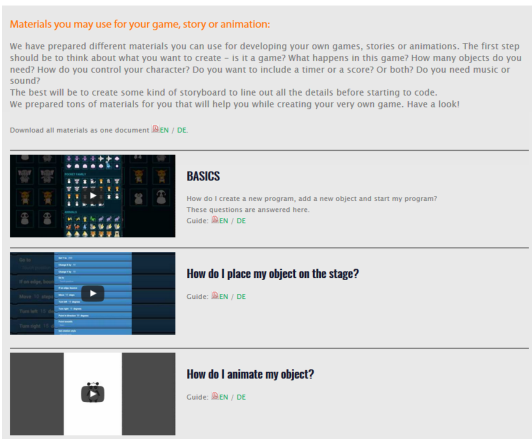


Figure 3.20.: New tutorial cards summer 2016.

As part of the GMTF, during NOLB several guides and templates have been developed, regularly updated, and maintained, which resulted in a number of 20 tutorial cards[75] (see Figure 3.20).

For the project, these tutorial cards could be downloaded as a PDF file and they are available in two languages, German and English. In addition, the Catrobat Mediawiki[76] was introduced and has been made accessible through the app: short press on a brick opens a menu with a help option which redirects the user to a Mediawiki. At this page, the user can find a description for bricks and short videos (the Mediawiki is at the time of writing available in German, Spanish, and English, but it is planned to open it up to allow editing by our users).

Finally, during the Second Cycle, dedicated space for NOLB tutorials was integrated in the education platform[77]. This pages offers a personalized space with information for students, teachers, and for parents (see Figure 3.21). A more detailed description is part of Delivery 4.2 (Spieler et al., 2017). Teachers, students, and parents can find tutorials, lessons plans, and a FAQ-section here.

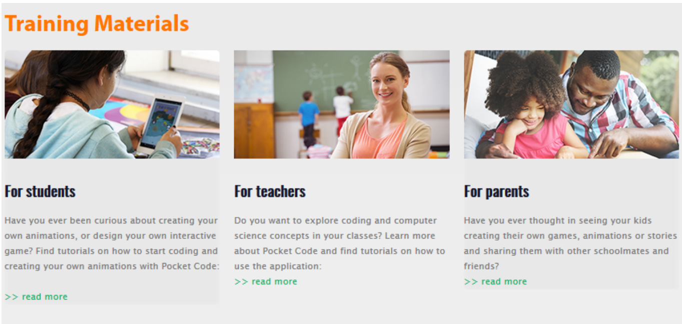


Figure 3.21.: NOLB on the educational platform.

3.4.1.1. Pocket Code game jams as a research method in NOLB.

The project has engendered the concept of game jams in school contexts. Game jam events have been organized with the aim of designing and creating small games in a short time-frame around a central theme, thus supporting engagement in informal learning situations beyond schools across a range of disciplines. The aim was to facilitate an exciting experience associated with strong, positive emotions, which should significantly support the achievement of learning goals. The goals for game jams during NOLB were to explore the use of that concepts as an integral element to the project. Therefore, the following sub-goals have been defined:

- explore game jams as a research method and to identify the advantages of this method;
- identify problems, such as difficulties in generalizing results and missing functionality in Pocket Code;
- introduce students to competitive approaches to increase motivation;
- evaluate participation and acknowledgement of participation as a reward mechanism;
- inspire meaningful learning by providing a unique interpretation and direction of academic content through personal game rules and selected game genre;

[75] Pocket Cards: https://edu.catrob.at/pocket-cards
[76] Brick documentation: https://wiki.catrob.at/index.php?title=Brick$_{Documentation}$
[77] No One Left Behind space: https://edu.catrob.at/no1leftbehind

- and create opportunities for official online game jams with Pocket Code.

To encourage schools to take part in the game jams, the project team organized online game jams through the Pocket Code community sharing platform. Via a community website, tutorials, media assets, and the game submission process was managed. The following information was provided:

- a general topic of the game jam event (announced ahead of the event);
- a surprising topic (announced shortly before the event to allow equal condition for all participants);
- diversifiers to increase engagement and spark creativity in the development process;
- landing pages with additional information, e.g., topic related support material (graphics, posters, and tutorials), and FAQ, etc.;
- project submission and a final questionnaire (integrated in the community website);
- and an overview of all submitted programs and a "Wall of Fame".

Additional aspects included:

- working in small groups (two to three students);
- integrating the "Shape of a Game" ceremony to encourage learners to use good design principles;
- creating awareness of license issues and attribution;
- and requesting to integrate an academic theme or objective in the game.

The framework for Pocket Code game jams is part of Delivery 4.2 (Spieler et al., 2017) (see Figure 3.22).

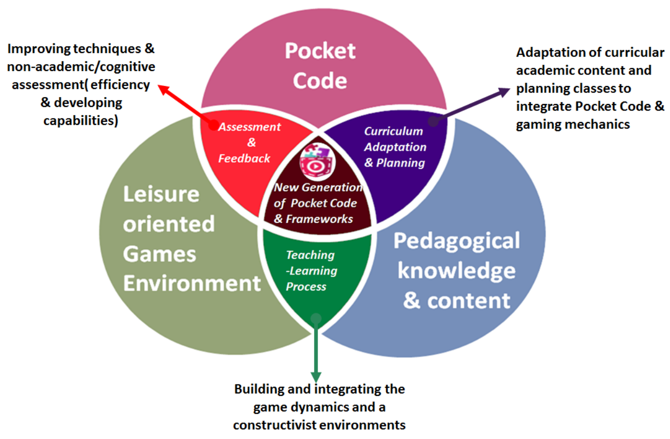


Figure 3.22.: NOLB game jam framework (Spieler et al., 2017).

The notion of game jams provides a paradigm for creating both formal and informal learning experiences, such as directed learning experience, problem-solving, hands-on projects, working collaboratively, and creative invention, within a learner-centered learning environment where children are creators of their own knowledge and learning material. For both performed game jams, the themes were chosen to reflect topics situated around subject disciplines where factors such as learning achievement, engagement, and persistence, as well as the development of computing and computational thinking skills, are important (Goddard et al., 2014). In each game jam, schools involved were encouraged to present the game jam event to students. The results of the two performed game jam events during NOLB are presented in the evaluation chapter (see Section 4.5).

3.4.2. New generation of Pocket Code.

The main results of the NOLB project included:

1. Create@School app, for the students' usage, and
2. Project Management Dashboard (PMD), for the teachers' usage.

New infrastructural services were developed, e.g., data tracking of certain events or certain times during game creation. These data was analyzed to create statistical information and visuals of students' behavioral parameters. Another important feature was the integration of different pre-coded game design templates and new features. Finally, the app also supports a set of accessibility features, which optimize the coding experience for students with special needs. The whole framework of Create@School is illustrated in Figure 3.23. All components of the new generation of the app are part of NOLB Delivery 4.2 (Spieler et al., 2017).

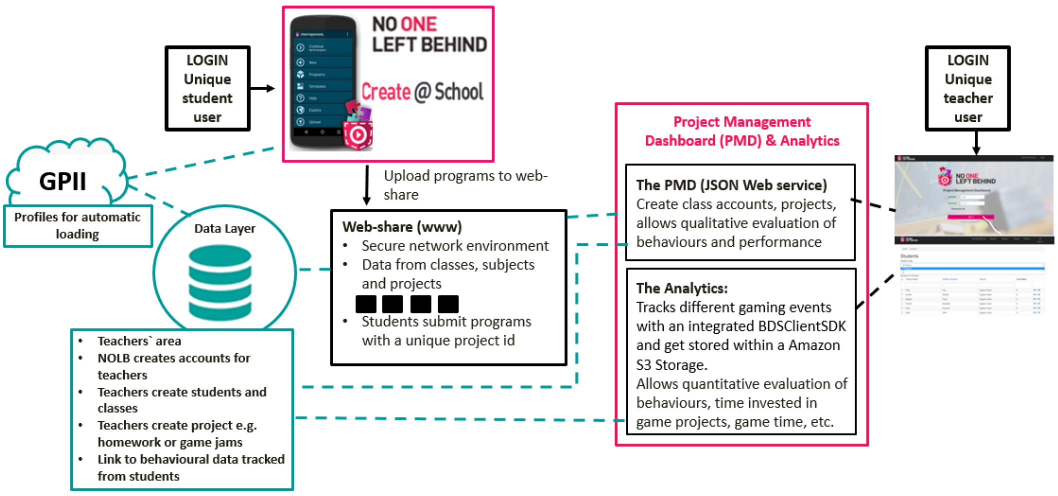


Figure 3.23.: NOLB framework for the new generation of Pocket Code (Collazos et al., 2017).

In this section, first, the components of the flavored version Create@School are explained. To limit the scope of this thesis, only the concept of the tools developed and evaluated at TU Graz are discussed in more detail, like the Create@School app as a whole, the data tracking, and the templates. Although the author was very integrated in the conceptualization and development of other parts, e.g., the accessibility preferences (developed as a whole at TU Graz), and the PMD (development of the interface to the community sharing platform), they are only described very briefly because they are not aligned with the aim of this thesis.

3.4.2.1. Create@School app.

The first version of Create@School was released on 17th of October 2016 as an open beta version on Google Play. The following Figures 3.25, 3.26, and 3.27 from Google Play provide a progress overview since the release.

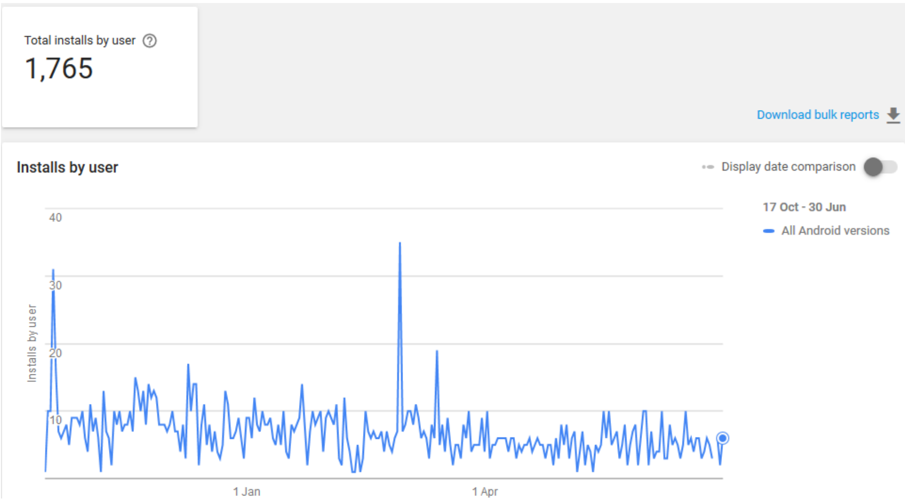


Figure 3.24.: Create@School-installs per user: The number of unique users who installed the app on one or more of their devices for the first time in the selected time range.

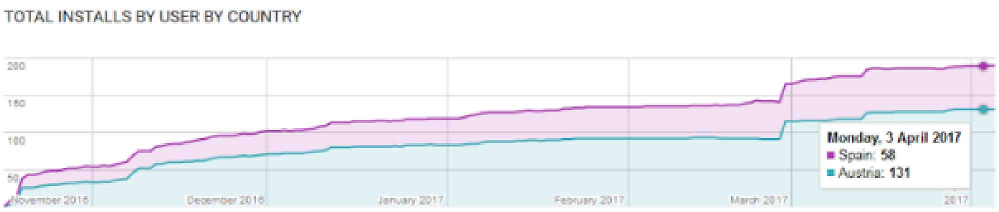


Figure 3.25.: Create@School: installs per country (Spain, Austria).

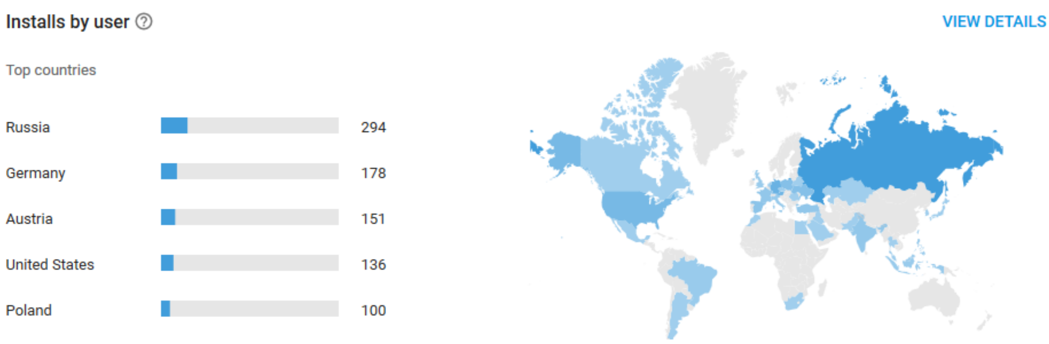


Figure 3.26.: Create@School: total installs per user.

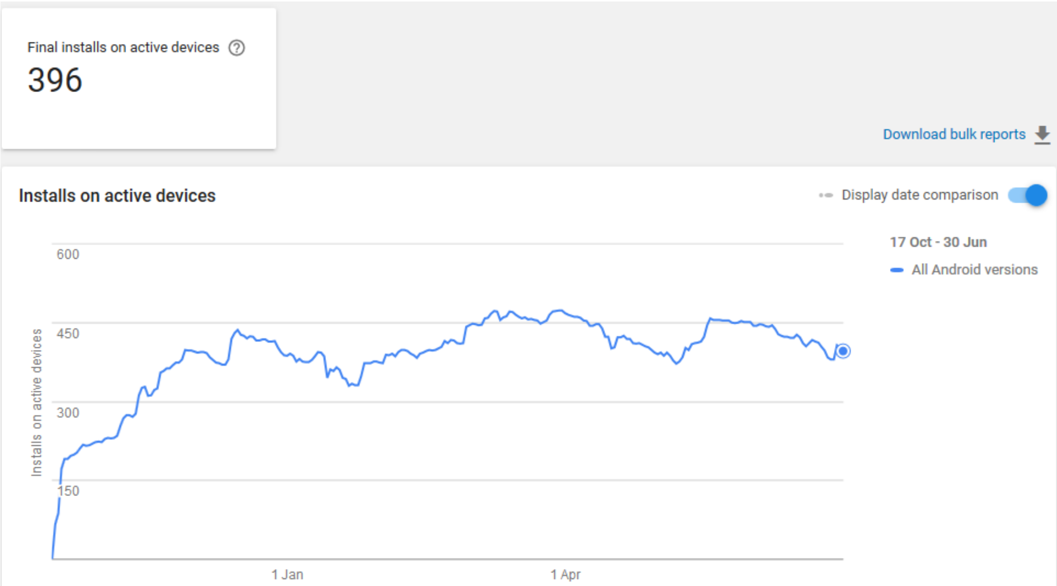


Figure 3.27.: Create@School-installs per device: The number of devices that have been active and on which the application is installed.

For the Second Cycle of the project, the first prototype of the Create@School app was introduced in the schools of the piloting countries (UK, Spain, and Austria). The app was improved and released seven times during the Second Cycle. With every iteration, Create@School got closer to a market ready application by including more translations, refactoring, templates, features and tracked events. Open beta testing (Google, 2018) allows researchers to run a test with a large group of users and to surface the app's beta version on the Play Store. When running an open beta test, anyone can join the beta program with a URL link and feedback can be submitted privately. Since Summer 2017 open beta apks are visible for all users within the Google Play Store but highlighted as beta. The (at this time private) URL link was shared with our teachers via email. Each tester needed to opt-in using the link to be part of the beta test (see Figure 3.28).

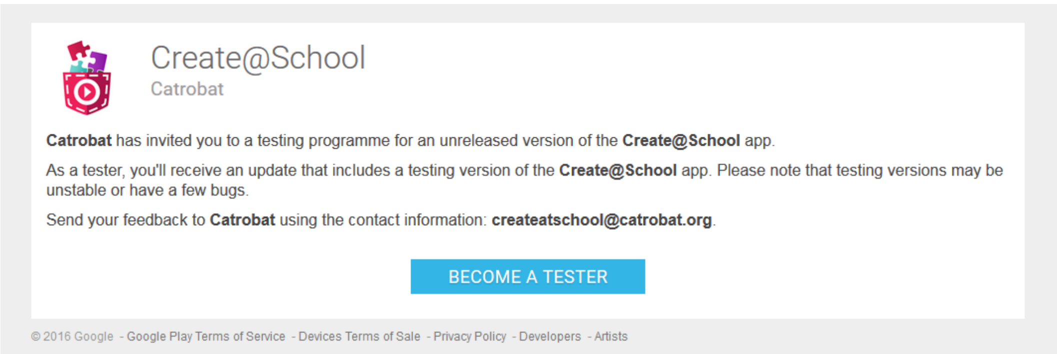


Figure 3.28.: Notification to become a tester for Create@School.

A login in order to use the app is required to bind the tracked data to one username. The NOLB team created 600 accounts per school with the following sequence of the username.

- nagif0001: n = NOLB, a = Austria, gi = school code, f/m for female and male and a sequential number

In addition, three teacher accounts per school have been created as well as test accounts for the NOLB team. This sequence ensures that the data of students is anonymized. To identify students' grade as well, every class started with a certain number, e.g., fifth grade 0001-0049, sixth grade 0050-0099 etc. Figure 3.29 shows the beta app in the Google Play Store.

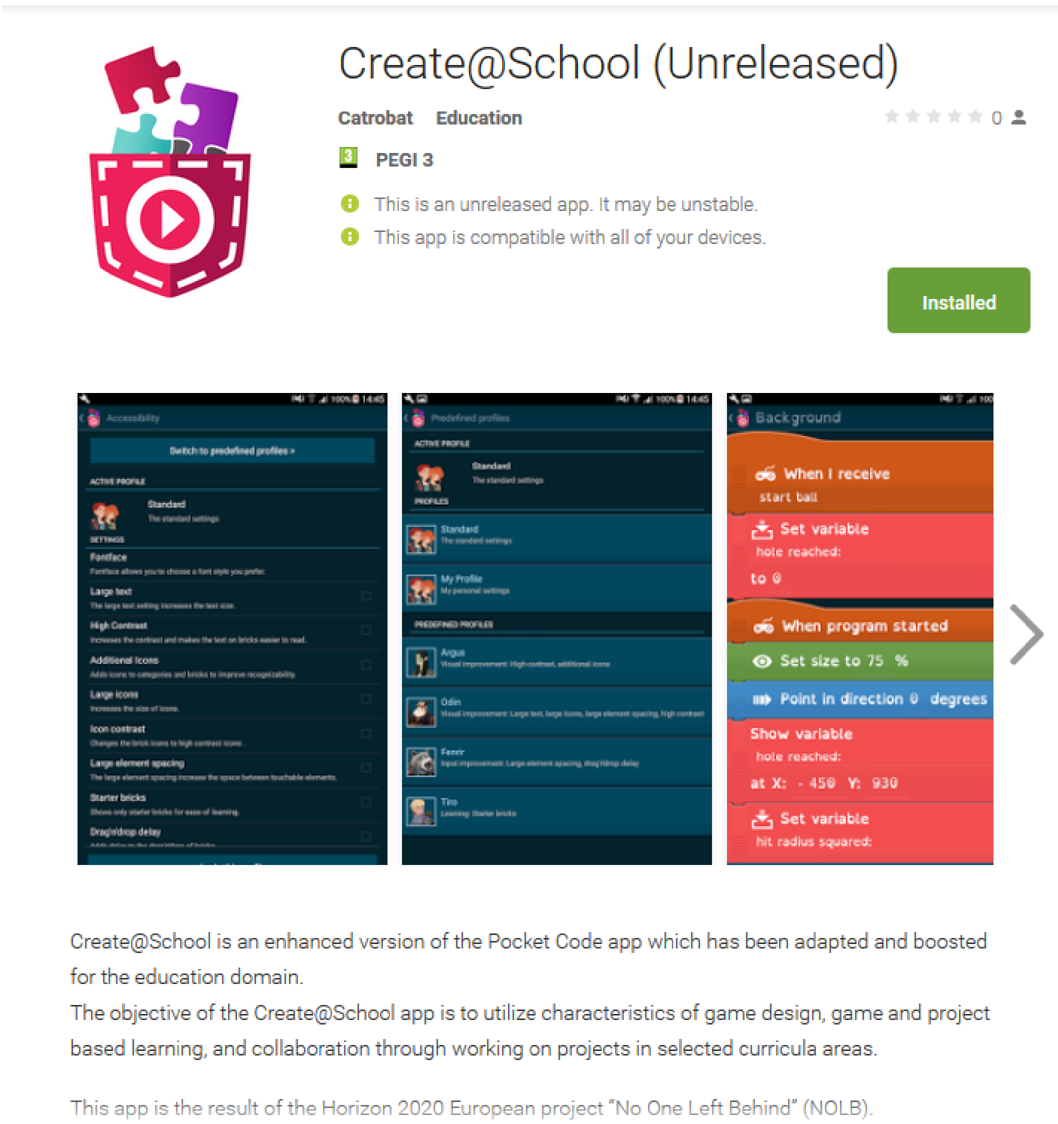


Figure 3.29.: Create@School at Google Play Store.

The objectives according to the NOLB proposal for the TU Graz team were the following:

- To customize Pocket Code towards a near market-ready version (New Generation of Pocket Code) by connecting the technology coming from gaming leisure oriented SMEs (methodologies and analytics engine) and inclusive GPII software.

At the beginning of the Second Cycle, all these elements were successfully integrated into the new app Create@School (see Figure 3.30). Thus, Create@School has been promoted as a flavored and enhanced version of Pocket Code for academic purposes to foster the needs of students, especially students who are often excluded in coding activities (girls, immigrants, and children with special needs). Flavoring, or the concept of multiple apk support allows for the publication of different APKs of the same app (Google Developers, 2018). Therefore, Create@School is a completely independent version of the application but shares the same package name and is signed with the same release key.

Figure 3.30.: Components of Create@School.

In May 2015, a list of 62 planned and forthcoming Pocket Code improvements were suggested. 76% of these features were integrated into the Create@School app and, as a result, the app was almost Scratch compliant. This was especially important because many teachers are already using Scratch at schools and Scratch is well-known among users all over the world (Meerbaum-Salant et al., 2010). A list of all implemented features is part of the Delivery 4.2 (Spieler et al., 2017). Other improvements and demanded features for the new app Create@School were the result of the usability evaluation of Section 4.3. Regarding the integration of predefined templates, a more powerful and usable interface for the management of large programs or a huge amount of objects has been developed. Therefore, two important features were integrated during the first project cycle: object grouping (to make the object view more concise and clearer), and the introduction of scenes (which eased the creation of large programs with more levels).

3.4.2.2. Pre-defined templates.

Create@School not only provides a newly developed game-like environment adapted to be used in school settings, but also templates to integrate game-based methodologies such as action, adventure, puzzle, and quiz. With this method, the team also strives to support female teenagers in the process of creating a game. Therefore, a focus group discussion was performed to collect the ideas of female teenagers for gamified templates that reinforce them especially (see Evaluation section). The game design and structure for the templates follow pedagogical principles (James et al., 2013). For example, the templates are a combination of entertainment and knowledge acquisition; they help with exploring content and solving problems, provide feedback to player, focus on essential learning contents, and stimulate the player to reflect on their experience, all of which are especially important for female teenagers (see Section 2.4.2) but also a promising way to encourage students of all genders.

Under this approach, the team have integrated ludic-oriented game mechanics which have been embedded through game templates, as well as rewards or victory points approaches. The templates enable game dynamics by editing an existing game design but allowing personalization of backgrounds, landscapes, characters, and the creation of new challenging levels at the same time, as well as changing the difficulty of a game (see important game elements in Section 2.2.1.1). The NOLB templates have been created by adapting game-modules, which are part of Delivery 2.1 (Boulton et al., 2016) (see Figure 3.31). These templates allow: 1) teachers to start with a well-structured program, and 2) students to add content and adjust the code to integrate their own ideas.

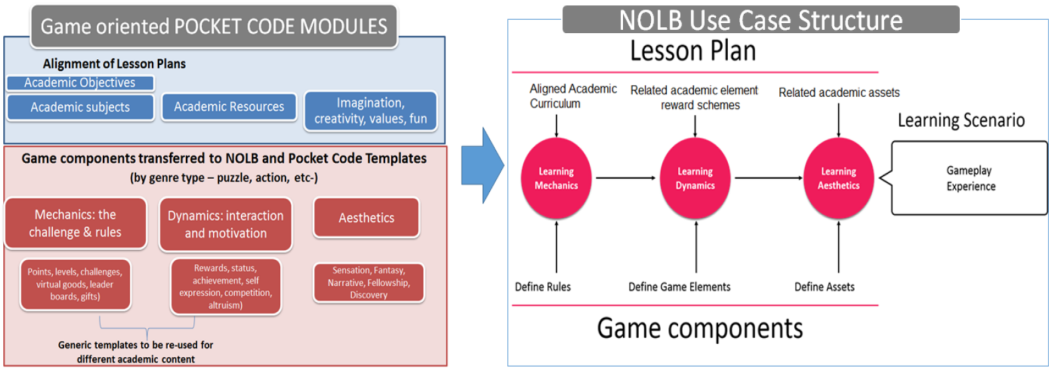


Figure 3.31.: Game oriented modules (Boulton et al., 2016).

The game-modules were developed based on different types of games (game genres; see Section 2.2.1.1). In the First Cycle, as planned, four game-modules were developed. For the Second Cycle nine more game-modules were developed (five since February 2017 and the last four since May 2017), based on sub-genres. The first four templates were directly stored in the app. To ensure a continuous extension, exchange, and improvement of the templates, and to avoid the app getting too big, we outsourced them in February 2017. They are now available for download directly in the app.

For integration of the templates into Create@School, a new option “Templates” has been added to the Create School’s main menu to access the templates list (see Figure 3.32; the screenshots also show the scenes and grouping feature — two last pictures). The developed templates have been described in Delivery 3.2 (Martinovs et al., 2017), Delivery 2.4 (Martinovs and Barrett, 2016), Delivery 4.2 (Spieler et al., 2017), and in the paper (Spieler et al., 2017).

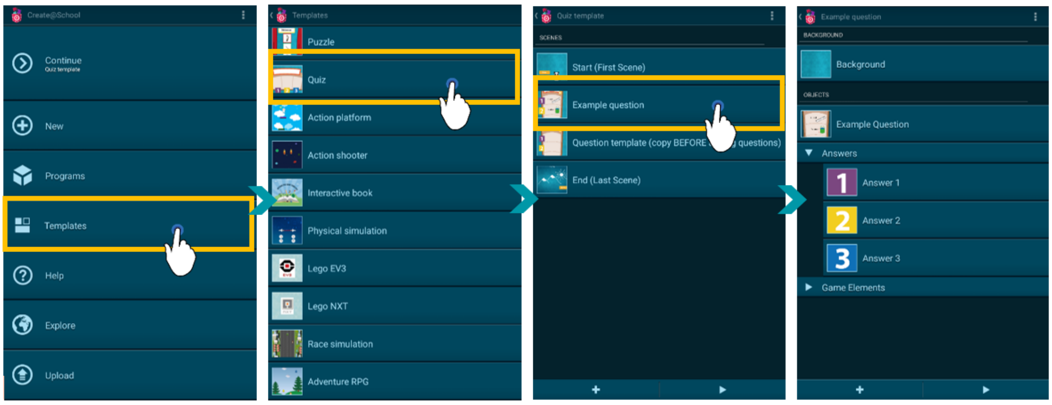


Figure 3.32.: Game templates in Create@School that supports scenes and grouping of objects.

All templates have important elements in common. They:

- are all based on the "Shape of a Game" (see Section 2.2.1.1), thus they start with a game screen (in different colors), followed by an instruction screen and end with end screen (see Figure 3.33);

Figure 3.33.: The "Shape of a Game" ceremony.

- contain an example level that fits to one theme/subject to show the user the gameplay, e.g. biology, music or physics;
- have a template level integrated that should be modified by the user;
- were available mostly in landscape and portrait mode;
- were translated in the languages of the participating countries;
- allow users to choose their own graphics and modify the "Shape of a game" — screens and play around with the different game elements;

- use different HUD (Poitschke et al., 2008) elements like points or a timer to reward users' actions;
- have at least two levels integrated which display different milestones that a player must achieve and hence provide a new challenge for them;
- have a logical structure;
- and are divided into different scenes and object groups.

Template tutorials for teachers and students were created (see the project's education platform). These guides represent a step-by-step tutorial and show how to use each template. They should help teachers to guide their students through the practice of game development while learning about academic subject content at the same time. The teacher then can decide which of the templates fits best to his or her subject and the learning goal within the curriculum.

The aim of the pre-coded game design templates was to support students during the game design process and help them to stick to good design principles. With this support, students' game design processes can be scaffolded, which allows them to focus on the game development itself. In addition, the templates allow learners to focus on subject-relevant problem solving activities rather than on understanding the functionality of the app. This leads directly to more time to express their creativity on different levels and more time for extra tasks. By using the game design elements, they can build new games and remix existing ones. First, the Tables 3.5, 3.6 and 3.7 provide a collection of all 13 pre-coded templates and second, Figure 3.34 shows a screenshot of every template. Three of the tempaltes are designed for a specific subject (template number 8, 9, and 10); the rest are designed to be applicable to various subjects.

Table 3.5.: Overview of the developed game templates 1 — 6. LG: learning goal. * beginner ** advanced *** expert.

1.*	Genre/theme: action/realistic MDA: points, challenge, reward, achievements
	Tutorial level: physics, properties of physical objects Gameplay: Tap on the correct answer (1, 2, or 3).
	To do: Provide questions/answers with text and images and add content per question. Adjust the template to the number of questions asked.
	LG: Add questions/answers/content to a specific topic, e.g., physics.
2.*	Genre/theme: action/realistic MDA: points, challenges, leaderboards, reward, achievements, competition, points
	Tutorial level: respiration, learn about de-/oxygenated blood cells Gameplay: Tap on correct objects avoid wrong ones (depending on the question).
	To do: Define an overal question. Add correct/wrong objects and adjust the variable “timer”.
	LG: Adjust the logic and add wrong/correct objects, e.g., blood cells/virus cells.
3.*	Genre/theme: puzzle/realistic MDA: points, challenges, reward, achievements
	Tutorial level: music, instrument groups Gameplay: Tap on the curtains to open them and to show the three available options. Tap on the one that does not fit with the others (“odd one out”). After 3 seconds the curtains close again and the score decreases.
	To do: Add looks and adjust the code (amount of puzzles), timer.
	LG: Logic challenge. Adjust the logic and add wrong/correct images, e.g., instrumental groups.
4. *	Genre/theme: text adventure/space MDA: levels, status, narrative, discovery
	Tutorial level: science, space Gameplay: Storytelling with linear choices. Listen to a question and decide "Yes" or "No". One question will always lead to the end of the game.
	To do: Add content and sound files to a specific topic and adjust the variable to define correct answers.
	LG: Add new scenes/animations e.g., for retelling a book or leading through a topic.
5.**	Genre/theme: action platformer/sky MDA: points, level, challenges, reward, status, achievements, sensation
	Tutorial level: physics, periodic system (halogens) Gameplay: Button-up scroller/doodle jump. Jump on the platforms to get points by catching the correct objects. Avoid wrong objects.
	To do: Define wrong/correct objects. Extra task: Add a level up.
	LG: Add looks and logic to emorize objects, e.g., halogens and non-halogens.
6.*	Genre/theme: action shooter/space MDA: levels, points, challenges, reward, status, achievements, fantasy
	Tutorial level: maths, division rules Gameplay: Shoot asteroids with the correct numbers/symbols.
	To do: Add looks with correct and wrong numbers/symbols and adjust the variable “timer”.
	LG: Add looks and logic to a specific topic, e.g., maths: devision of numbers.

Table 3.6.: Overview of the developed game templates 7 — 11. LG: learning goal. * beginner ** advanced *** expert.

7. *	Genre/theme: text adventure (interactive book)/realistic levels, self-expression, status, narrative, expression
	Tutorial level: science/water cycle Gameplay: An easy version of an interactive book by switching between scenes.
	To do: Add animations and descriptions. Create your own story
	LG: Add texts and graphics/animations. Add more scenes and retell a story/topic, e.g., the water cycle
8. **	Genre/theme: simulation/realistic MDA: levels, challenges, status, achievements, discovery
	Fixed subject: computer science/robotic Gameplay: Use the Lego NXT/EV3 extension and solve three levels.
	To do: Create a gamepad to control the robot, use the inclination sensors to control the robot. Create a maze by using the infrared sensor of the robot.
	LG: Add sensor values, and use coordinates to control a Lego NXT/EV3 robot.
9. **	Genre/theme: simulation/space MDA: levels, challenges, status, achievements, discovery
	Fixed subject: physics, Newton's laws of motion Gameplay: Perform physical experiments with Newton's 2nd law of motion and experiments with the formula.
	To do: Change the variable "force" by tapping a rocket. Calculate the acceleration. Add an own object and change its mass and acceleration. Extra task: Newton's 3rd law of motion
	LG: Repetition Newton's 2nd and 3rd law and physical properties.
10. ***	Genre/theme: sumulation/comic MDA: points, levels, challenges, reward, status, achievements, gifting, altruism, narrative
	Fixed subject: life after school/work skills Gameplay: Help the character to get ready for work and to serve the correct order and keep the customers happy (2 independent levels).
	To do: Level 1-Change the sequence of the actions according to your own morning routine. level 2-Change the workplace, e.g., different items to order (change items, speech bubble and sound effects).
	LG: Add logic/looks and to be able to follow/change sequences or to identify equipment needed to. Complete a range of jobs
11. ***	Genre/theme: strategy/realistic MDA: points, levels, challenges, reward, status, achievements, sensation, submission
	Tutorial level: history/important historical persons Gameplay: Multiplayer (on one device). Strategy decisions via a simple mile template and chronical orders.
	To do: Add logic for a mile for a 6x6 gameboard. Add coins with symbols related to the topic.
	LG: Define items and set them in a chronical order, e.g., important historical persons

Table 3.7.: Overview of the developed game templates 12 — 13. LG: learning goal. * beginner ** advanced *** expert.

12. *	Genre/theme: racing/comic MDA: points, levels, challenge, reward, status, achievement, sensation
	Tutorial level: science/pollution Gameplay: Move the car by tilting the phone. Collect items to get points/level up. Avoid the other cars to keep playing.
	To do: Add objects to increase the difficulty of the game and add more levels.
	LG: Add looks and logic of specific topic, e.g., pollution.
13. ***	Genre/theme: adventure RPG/nature MDA: points, levels, challenge, reward, status, achievement, sensation
	Tutorial level: fine arts/color circle Gameplay: Create your own character and choose between different players. Choose a project and collect items you need. Collected items are part of your inventory. You can drop them again. Finalize the project by mixing together the different parts.
	To do: Add your own project and items to collect and later to merge, mix, or build parts
	LG: Add looks and logic to collect and merge/mix/build parts., e.g., draw a painting by mixing colors.

Figure 3.34.: Templates screenshots 1-13.

In Austria, two templates (physical simulation and adventure RPG) were evaluated (see Section 4.3.5).

3.4.2.3. Accessibility preferences.

The goal of integrating GPII into Create@School was to make the IDE more usable for various user groups. Software programs, mobile apps, and websites have a default user interface tries to cater to many people, but that is often unsuitable for people with special needs. Many such programs are adaptable, but on-site observations have shown that most people never adapt the settings of the software they use. This may be because they are afraid of breaking something, or because it is too difficult. Another reason is that devices at schools are not used every lesson by the same student. Thus, from a functional perspective, Create@School allows the personalization of the game-playing platform through profiles, as students can choose a preferred profile in Create@School whenever they want. The chosen profile only influences the Pocket Code app and not the general device settings. Features integrated to adapt the UI for users with special needs compromise: Choose Font Face, Larger Text, High Contrast, Additional Icons, Large Icons, Icon Contrast, Large Element Spacing, Starter Bricks, Drag'n'Drop Delay, and Hints and Tips. The profile-changing option has been embedded within the Create@School menu, and made profiles selectable under "a name" (to make them more distinctive and attractive for the user). The profiles have the names of gods to make profiles unique, attractive, and to avoid discrimination. Each profile is linked to a special needs usage. Therefore, Argus is a profile for visual (color) impairment, Odin for general visual impairment, Fenrir for motoric impairments, and Tiro for a beginner mode. The first prototype of this feature was released as part of the Create@School app for the Second Cycle. Figure 3.35 illustrates the possible changes of preferences in the Create@School UI. The accessibility preferences are part of Delivery 4.2 (Spieler et al., 2017).

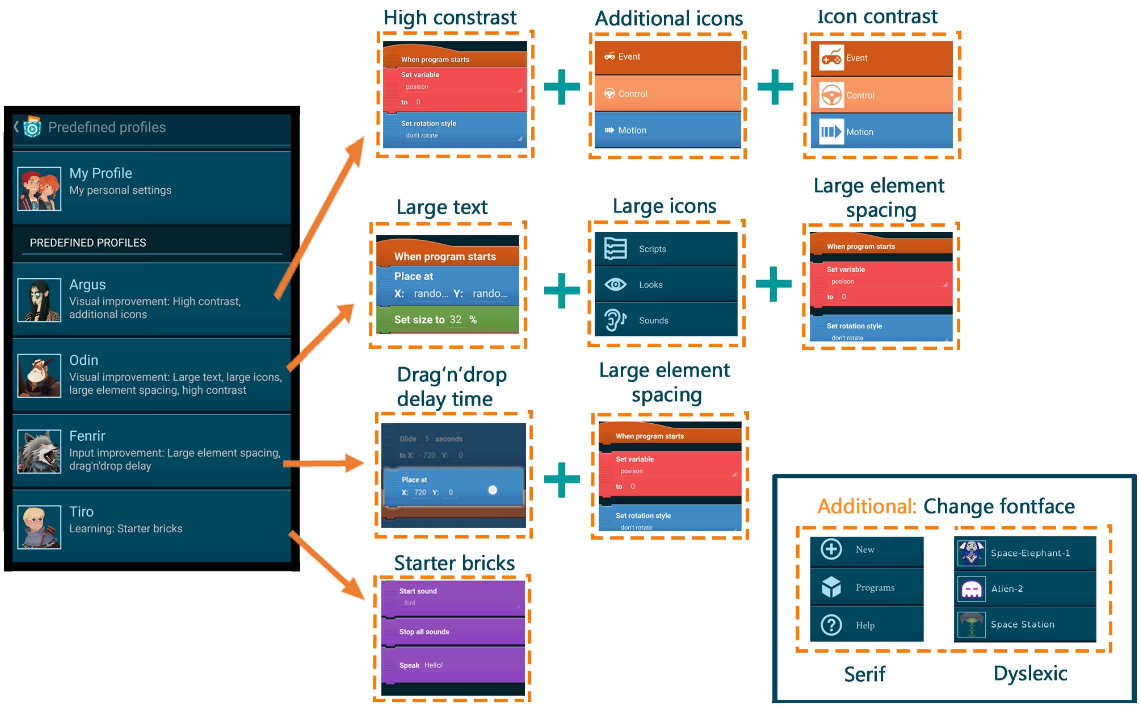


Figure 3.35.: Accessibility settings and predefined profiles.

For novice users, three more features have been implemented. First, the display of hints, second, help options for bricks (linked to the Mediawiki), and an intro for the formula editor (as a result of Section 4.2.3). In Austria, students were not explicitly encouraged to use these options.

3.4.2.4. Tracking of analytics data.

To help teachers in the assessment and feedback process of students' projects — or the *why* of learning (see UDL) — analytics data was tracked to provide visualizations. As a result, the "NOLB Behaviour Measurements" were generated (see Delivery 3.3 (Collazos et al., 2016) and Delivery 4.3 (Collazos et al., 2017)).

First, different types of events were tracked by integrating an SDK into the source code of Create @School which performs the actions of "writing" or "registering" whatever actions the user performs while using Create@School. Tracked actions included creating a new program, adding a brick, time spent with programming, or the use of game templates. All these events are sent via JSON and stored in a database. For this the app used the commercial BDSClientSDK (Big Data Services Client software development kit), which is a very simple to use and lightweight library without external dependencies that allows developers to send different types of events related with their applications to the Big Data Services (BDS) platform. In that way, it is possible to explore information about users, their sessions, and their actions. This SDK was provided by the Spanish NOLB partner ZED and integrated as a whole by TU Graz. Therefore, the events and parameters were added to the source code (e.g., event name: "createProgram" with the parameters program name, landscape-true/false, example program-true/false, etc.) to send it to the BDS. The SDK allows the definition three kinds of events: "custom", "init", and "end" events. Mostly custom events were used because they allow to add details, e.g., brick name, or brick category. The "init" event is defined to start a session (user logs in) and the "end" event for closing the session (user logs out). Altogether, approximately 90 events were integrated from TU Graz into the source code. Second, for defining the behavioral dimensions linked to the learning process, a bottom-up approach has been used. This process started from the lowest level of coding, where all coding bricks and performed events from the Create@School app were listed. Tracked events were comprised in the categories "event creation" for add or edit events, "event deletion" for deletion events, "coding skills" that differs between basic and advanced bricks or functionalities, "look and feel customization" for customization of assets, use of personal resources, and enabling of accessibility settings, "time management" for the tracking of different times during coding (e.g., time spent with playing/test, with research, etc.), and "support" for using help options like hints. The results of these behavioral measurements are part of the next chapter (see Section 4.3.2).

3.4.2.5. Project Management Dashboard (PMD).

With the *Project Management Dashboard* (PMD)[78], teachers are able to assess students' programs. Therefore, the PMD provides the general framework needed to manage students and classes for each teacher of the NOLB project. Through the PMD, teachers can plan, assign, and manage the delivery of game projects to support new game-based teaching approaches. Therefore, students upload their games to the default community sharing platform. To submit the programs to the PMD, a new button was integrated into the detail page of the sharing program with the name "Submit your project". If the student clicks on this button, he/she must enter a unique id(provided by the teacher) which links the submitted program to the corresponding project/student in the PMD which allows assessment by the

[78]NOLB PMD: https://www.pmdnolb.cloud/index.php

teacher. Figure 3.36 holds an overview of the PMDs interface. The description of all components of the PMD is part of Delivery 4.2 (Spieler et al., 2017) and Delivery 4.3 (Collazos et al., 2017).

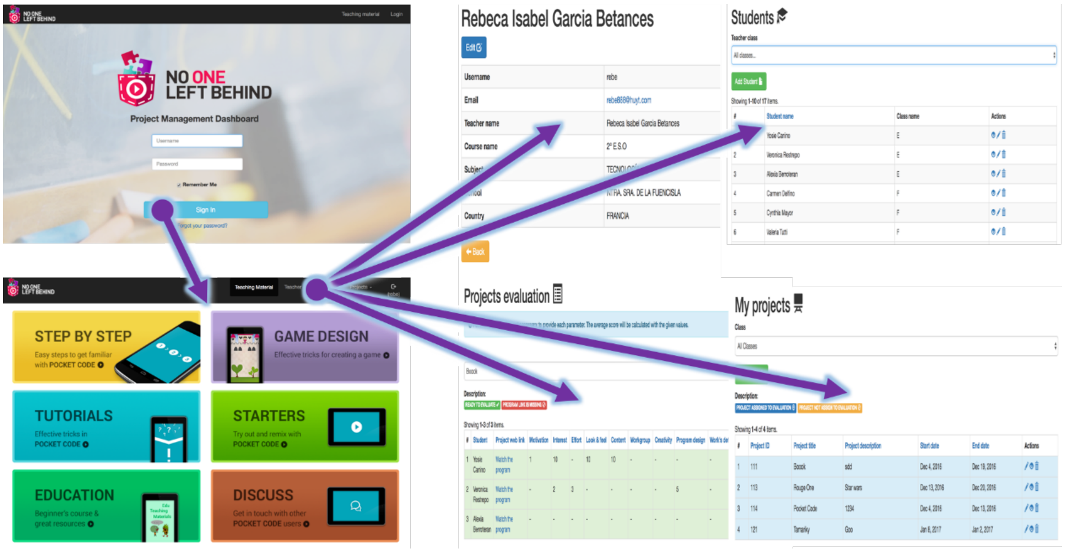


Figure 3.36.: PMD interface (Spieler et al., 2017).

Another primary role for the PMD was to allow quantitative and qualitative evaluation of students regarding their completion of projects and achievements of objectives. The information observed by teachers captures the perception of them regarding the academic readiness, achievement of learning goals, and students' abilities or social behaviors. To link these parameters to the learning process, again a bottom-up approach is used. Different types of behaviors that affect the learning process have been gathered during the First Cycle of the project and a set of behavioral constructs has been selected and customized by the UK team at NTU. As a result, seven observational parameters have been defined: matching of the academic and gaming objectives, game originality, functionality of the game, presentation of the work, collaboration, and amount of teacher interventions. Teachers defined these parameters for each student (see Figure 3.37).

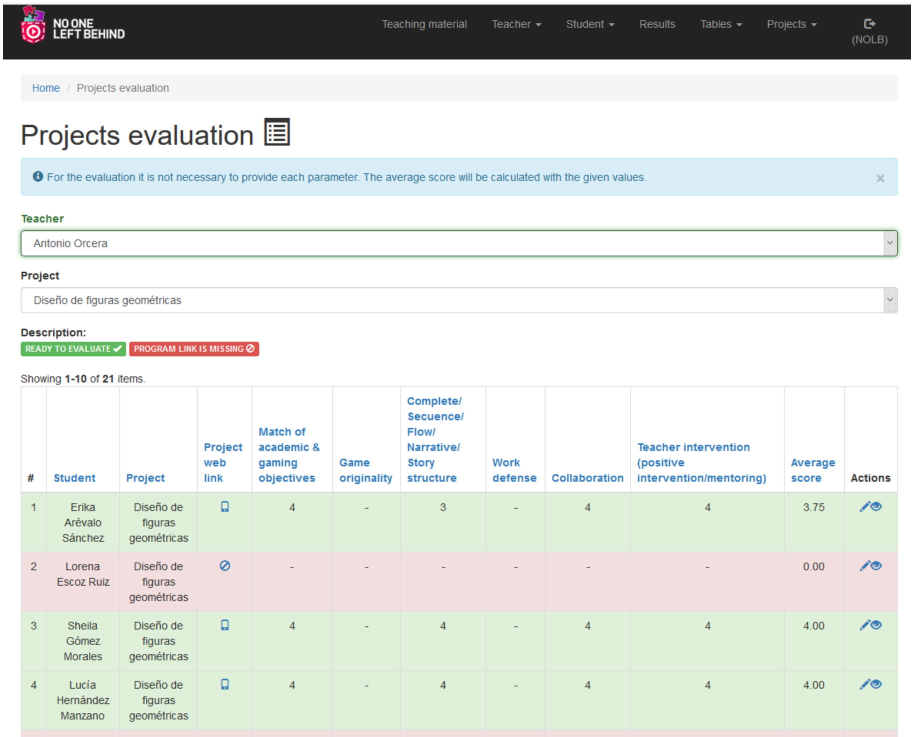


Figure 3.37.: Projects evaluation: provides an efficient way to test programs (web-player on the sharing platform) and to evaluate the student's work. Example from Spain.

3.4.2.6. Analytic dashboard: visualization of cognitive and behavioral measurements.

The *Analytic Dashboards* link observational data coming from the PMD, with the automatically gathered data tracked with the SDK and stored in the BDS. Thus, it links the top-down and bottom-up approach described in the previous section. All data was stored in a data lake, which allows different kinds of analyses based on pre-defined behavioral constructs. The overall architecture of the Analytics Engine is presented in Figure 3.38.

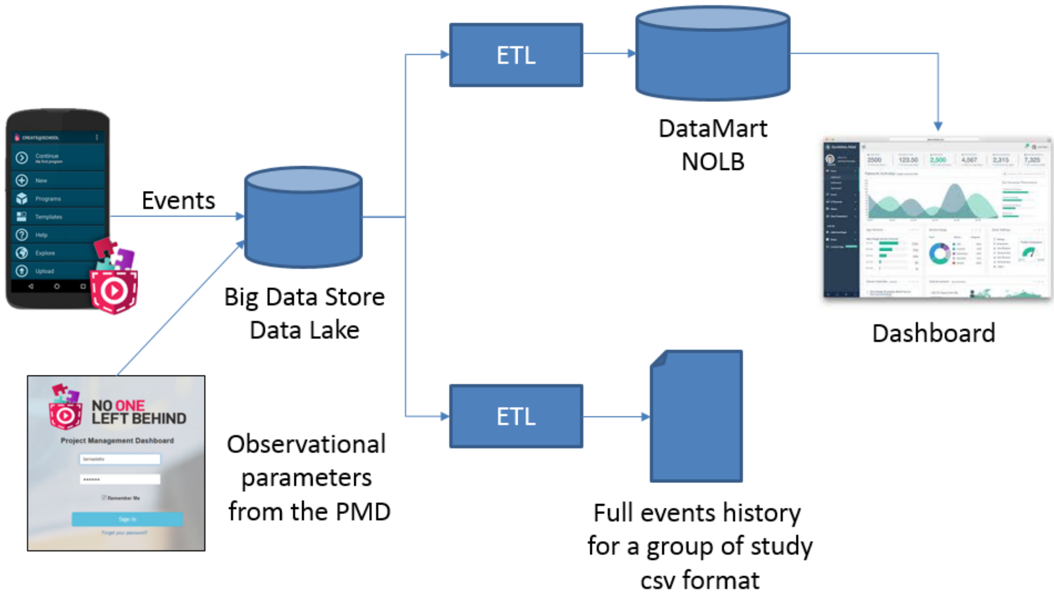


Figure 3.38.: Analytics Engine: architecture (Collazos et al., 2017).

All constructs (data from Create@School and the PMD) were categorized in four types: didactical/cognitive, socio-behavioral, self-regulatory, and engagement. Each construct has been defined through a set of factors (categories) that build or link to a set of behaviors that support learning. The following table 4.4 shows the schema of the constructs and its related behavioral factors (defined by the UK based team). Some of them, like confidence, creativity, interest, and self-efficacy are highly relevant socio behavioral factors for female teenagers (see Section 2.4.2).

Table 3.8.: The behavioral constructs and related behavioral factors (Collazos et al., 2017).

Didactical/Cognitive	**Socio Behavioral**	
• Academic Readiness • Usage of Create@School • Coding	• Confidence • Creativity •Interest	• Performance • Positive Affect • Self-Efficacy
Self-Regulatory	**Engagement**	
•Effort/Dedicated Time • High-Thinking • Persistence	•Action-Engagement • Self-Engagement (over average in positive affect) • Social Engagement	

The following Figure 3.39 presents the behavioral oriented matrix example. Each one of the parameter from Table 4.4 is the result of the underlying behavioral factors.

Behavioural constructs (bottom-up)	Match of academic & gaming objectives	Game originality	Complete/sequence/flow narrative/story \structure	Work defence	Collaboration	Event Creation	Event Deletion	Coding Skills	Look & Feel/Customi-sation/Aesthetics	Customizes/Changes GPII	Absolute time spent	Tracked events/actions per minutes	Time spent w/playing-testing the game	Time spent in Pocket Paint	Time in Web View	Time spent in research/tutorials	Time spent in program-ming (remaining time)	uses brick help option and hints	Teacher intervention (positive intervention / mentoring)
Gathering data process																			
Observational: via PMD	**Teachers` observation**																		**Support (observation)**
Automatic: via SDK						**App related**					**Time Management (app related)**							**Support (App)**	
Behavioural constructs (top-down)																			
Didactical/Cognitive																			
Academic Readiness	x		x	x															
Usage of Create@School						x	x	x											
Social Behavioural																			
Confidence	x			x		x		x											
Self-efficacy	x							x			x	x							
Performance	x		x					x									x		
Positive Affect	x	x							x	+				x					x
Interest	x					x			x					x	x	x	x	x	x
Creativity		x	x						x										
Self-Regulatory																			
High Thinking	x		x					x											
Persistence							x								x	x	x	x	x
Effort/dedicated time												x	x	x	x	x	x		
Engagement																			
Self-engagement									x	x			x	x					x
Action-Engagement						x		x	x										
Social Engagement					x														

Figure 3.39.: Create@School behavioral matrix (Collazos et al., 2017).

All these data undergo a "transformation" process, following a standardized methodology (gaussian/normalization methodology (Roell, 2017)) for dynamically customizing the developed dashboards. These dashboards can then be navigated using several dimensions, like students (either individually or as part of the class), and the actions they have performed while using Create@School. A one to four scale was used to reflect the performance of each student in each sub-category and category (1: entry level, 2: developing, 3: secure, 4: mastery). Two types of visualization dashboards have been provided: online (big data software -Tableau[79] — supported) and offline (excel sheets). Both dashboards could dynamically update the different behavioral constructs or categories when selecting different variables, such as teacher, the class or a student. These dashboards are part of the results chapter (see Section 4.3.2) and part of the Delivery 4.3 (Collazos et al., 2017). However, considering and discussing these factors in more detail is beyond the scope of this thesis. Missing details will be added where they are needed within Chapter 5 by presenting the results of the behavioral measurements from the Austrian pilots.

3.5. NOLB: The Austrian Pilot Plan

The Austrian team developed different services and tools (see previous chapter) but also conducted classroom courses at the pilot schools. At the beginning of NOLB, the Austrian team consisted of two

[79]Tableau ZED: https://tableau.zed.com

women and two men. During the whole period many students (developers, educators, or designers) helped with NOLB-related issues and the team changed constantly in its constellations but was mostly gender equal. This section explains the setting of the courses in Austria, the approaches used in the classroom, and the different projects performed with Pocket Code and Create@School. Finally, a summary will be provided that shows all actions that have been completed in order to reinforce female teenagers in schools.

3.5.1. Setting of the NOLB coding courses.

For NOLB, the coding classes had all a constructionist base in common (see Section 2.1.3.1). Thus, students were actively involved in the process of constructing a game, they took charge of their own learning, and found solutions by using the tutorial cards, asking their peers, or just trying out alternatives and learning from their mistakes. Therefore, students learned to be patient, felt ownership of their achievements, and truly understood many of the basic programming steps. For the purpose of designing and creating artworks for backgrounds and objects, students were provided extra time during additional arts lessons (which seemed to be important for girls, see next section). The setup of the school units was very similar through all countries. In Austria, one unit had a duration of 50 minutes and the coding courses were conducted mostly in double units. First, an instruction unit was provided to introduce the project, the team and the app itself, with example games and an explanation/demonstration of the UI. Instruction units were followed by two or four double session units where students worked on their specific project (the main learning) and where students programed their own games or made extensions to existing games (templates). In many cases, students started with drawing a storyboard to have a clear image of their game and to identify the needed game elements, characters and programming concepts (see Section 2.2.1.1). In the last unit, students could present their games in front of their peers (in plenary), followed by a discussion round. Depending on the available units, the age of students, and their previous programming experience, three different learning approaches were used; these approaches have been already presented in the paper (Petri et al., 2016).

Approach 1 (A1)/Providing a Framework/Template: Students of lower grades (Grade 6 and 7, and classes with less than four units) got a framework (FS/FC) or a pre-coded template (SC) in which specific parts in the code were missing (indicated by a note brick), e.g., collision detection or using inclination sensors, see Figur 3.40. Within small groups, the team developed this missing functionality together with them to guarantee a student-centered classroom, where students' learning and discovery was in their own hands.

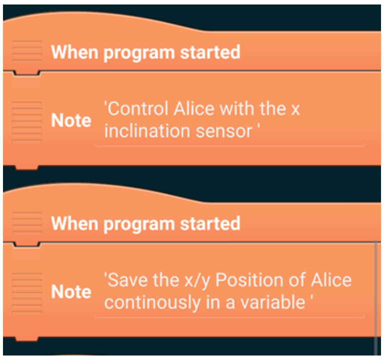


Figure 3.40.: Approach 1 (A1)/Providing a framework/template: Specific parts in the code are missing indicated by a "Note"- brick.

Approach 2 (A2)/Learning-by-doing: Students from higher grades (up to Grade 9) with at least six units got a short introduction unit including a short hands-on sessions guided by the team and by using tutorial cards. In these sessions, one student at one time was asked to come to the front of the class and tried to add one small but meaningful new feature/functionality to the game (e.g., let the object appear randomly on the stage, see Figure 3.41). If the student had problems with this task, the rest of the class could help him or her out. At the end, the whole class programmed a whole game collaboratively by learning the main functionalities of the app.

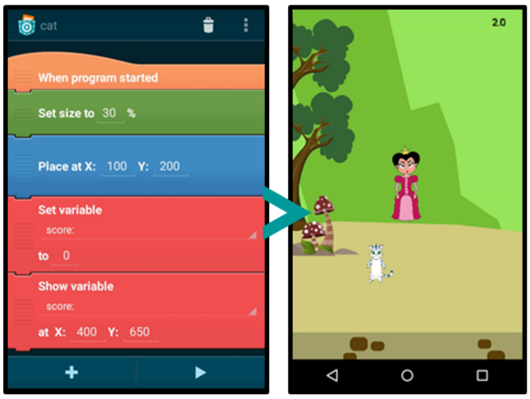


Figure 3.41.: Approach 2 (A2)/Learning-by-doing: guided by the team and tutorial cards.

Approach 3 (A3)/Hands-on: In some classes students used a more learning-by-doing approach. In these classes, the bricks were printed out on paper and used to form scripts (i.e. physical flashcards). Thus, students were trying to predict the outcome of their games beforehand (before they actually had used Pocket Code on a mobile device). They could freely choose what kind of games they wanted to program and got only minimal guidance. An example of bricks on paper can be seen in Figure 3.42.

Figure 3.42.: Approach 3/Hands-on: bricks on paper (flashcards).

The NOLB team attended all units to make notes about observed problems, gender relevant differences, bugs or difficulties of the app itself. Pre- and post-questionnaires, observations, documentations, video, and picture material was collected during all units.

3.5.2. Performed NOLB coding courses.

Altogether, 22 coding courses were conducted in Austria during NOLB, and the apps were integrated in the curricula of English, computer science, physics, fine arts, and music. A description of the courses was part of the author's paper (Spieler et al., 2017). Teachers who had introduced this gamified approach during the Feasibility Study (FS) and the First Cycle (FC) used the app Pocket Code. Teachers who had classes during the Second Cycle (SC) used the app Create@School. During FS, eleven courses were performed; during the FC, two courses; and during the SC, nine courses. Teachers could decide whether students should: work in small groups (SG) on a joint program where every student would then create one game level which at the end would be merged into one big game; do pair work (PW), where two students would work on the same program (either at one or at two tablets); or work individually (I). In addition, students were able to ask the NOLB team for assistance during their lessons, e.g., questions specific to Pocket Code. The courses per pilot school are visualized in Tables 4.5, 4.6, and 4.7. In addition, at pilot school 3 Teacher 11 provided again during the SC, accounts for the students of course 20 (voluntary course). A1-A3 refer to the used approach, see previous section.

Table 3.9.: Courses at pilot school 1: GIBS. LG: learning goal.

Courses during the Feasibility Study (FS)	
Course no. 1,2	Used approach/learning scenario: A2/SG Teacher no.:1, 2, 3, 4 — 47 students (two classes, f: 32, m: 17), Grade 9, 6 units Subject/topic: English and computer science/book retelling LG: To program an interactive quiz game in computing to a book read in English class.
Course no. 3	Used approach/learning scenario: A3/I Teacher no.: 4 — 24 students (f: 13, m: 11), Grade 9, 10 units Subject/topic: computer science/game design LG: Create a simple game (start, instruction, game, and end screen).
Course no. 4	Used approach/learning scenario: A1/PW Teacher no.: 5 — 16 students (f: 17, m: 9), Grade 7, 8 units Subject/topic: physics/density of objects and liquids LG: Create a game where objects glide according to their physical properties (density) and applying the formula.
Course no. 5	Used approach/learning scenario: A3/SG Teacher no.: 6 — 24 students (f: 16, m: 8), Grade 9, 8 units Subject/topic: fine arts/game design LG: Create a game and finish one level per student.
Course no. 6	Used approach/learning scenario: A3/S Teacher no.: 6 — 25 students (f: 12, m: 13), Grade 10, 8 units Subject/topic: fine arts/game design LG: Create a game and finish one level per student.
Courses during Second Cycle (SC)	
Course no. 7-9	Used approach/learning scenario: A3/SG Teacher no.: 6 — 74 students (3 classes; f: 47, m: 28), Grade 9, 4 units Subject/topic: fine arts/game design LG: Use the “Shape of a Game ”, avoid/catch something, or tell a story and add an interactive part.
Course no. 10	Used approach/learning scenario: A1/I Teacher no.: 5 — 25 students (f: 17, m: 8), Grade 7, 4 units Subject/topic: physics/Newton’s 2nd law of motion LG: Create a game where objects glide according to their physical properties (mass, acceleration) and apply the formula. Used template: Physical simulation
Course no. 11	Used approach/learning scenario: A1/I Teacher no.: 4 — 12 students (f: 6, m: 6), Grade 9, 10 units Subject/topic: computer science/game design LG: Create an adventure RPG game and apply it to different subject areas e.g., biology, music, etc. Used template: Adventure RPG

Table 3.10.: Courses of pilot school 3: Akademisches Gymnasium. LG: learning goal.

Courses during the Feasibility Study (FS)	
Course no. 12	Used approach/learning scenario: A1/PW Teacher no.:8 — 29 students (f: 17, m: 12), Grade 6, 8 units Subject/topic: fine arts/Alice in Wonderland LG: Create a vocabulary game by adding missing parts within the code.
Course no. 13	Used approach/learning scenario: A2/SG Teacher no.: 7 — 21 students (f: 15, m: 6), Grade 9, 8 units Subject/topic:computer science/game design LG: Create a game that include the "Shape of a Game ".
Courses during First Cycle (FC)	
Course no. 14	Used approach/learning scenario: A2/I Teacher no.: 7 — 11 students (f: 6, m: 5), Grade 9, 8 units Subject/topic: computer science/pyhsics quiz LG: Create a physics quiz template for Grade 6 (prework for quiz template).
Course no. 15	Used approach/learning scenario: A1/PW and I Teacher no.: 7 — 29 students (f: 17, m: 12), Grade 6, 8 units Subject/topic: physics/structure of matter LG: Add five questions to the quiz; Used template: Quiz
Courses during Second Cycle (SC)	
Course no. 16	Used approach/learning scenario: A1/PW and I Teacher no.: 8 — 29 students (f: 17, m: 11), Grade 7, 4 units Subject/topic: fine arts/renaissance, baroque, and romanesque LG: Create a puzzle game with 5 levels and add your own graphics. Used template: Puzzle
Course no. 17	Used approach/learning scenario: A2/SG Teacher no.: 8 — 11 students (f: 6, m: 5), Grade 11/12, 3 units Subject/topic: computer science/game design (voluntary) LG: Create an adventure/action game (start, game, end screen).
Course no. 18	Used approach/learning scenario: A1/I Teacher no.: 8 — 11 students (f: 6, m: 5), Grade 11/12, 3 units Subject/topic: computer science/game design (voluntary) LG: Create a quiz with 5 questions; Used template: Quiz
Course no. 19	Used approach/learning scenario: A1/I Teacher no.: 7 — 29 students (f: 17, m: 11), Grade 7, 3 units Subject/topic: physics, physical experiments. LG: Add an explanation and animation of a performed experiment. Used template: interactive book (text adventure)

Table 3.11.: Courses of pilot school 3: Borg Birkfeld. LG: learning goal.

Courses during the Feasibility Study (FS)	
Course no. 20	Used approach/learning scenario: A1/PW Teacher no.: 9 — 13 students (f: 5, m: 8), Grade 9, 4 units Subject/topic: computer science/computing quiz LG: Use sound to record the questions and use images for the answers.
Course no. 21	Used approach/learning scenario: A1/PW Teacher no.: 10 — 21 students (f: 16, m: 5), Grade 9, 4 units Subject/topic: music, muscial instrumts quiz LG: Add sounds and catch the corresponding musical instrument.
Courses during Second Cycle (SC)	
Course no. 22	Used approach/learning scenario: A2/PW and I Teacher no.: 11 — 19 students (f: -, m: 19), Grade 9, 4 units Subject/topic: computer science/Galaxy (participated in the Galaxy Game Jam, voluntary course LG: Create an action game (start, game, and end screen) to meet the topic galaxy

3.5.3. Gender inclusiveness in NOLB.

To address the gender bias in coding classes, the goal of the Austrian study included how to make Pocket Code more accessible and attractive for female teenagers (Delivery 5.4 (Spieler and Mashkina, 2017)). Although teenage girls constitute a large group that play smartphone and tablet games (Melton, 2017), turning them from mere consumers to active creators of games is a challenging task. The goal was to ensure a positive first experience in programming for girls with the education apps, which may direct their future career choice towards STEM fields. The assumption is, that it is possible to spark girls' interests by getting them engaged in computational thinking (see Section 2.1.4) through collaborative, creative, and engaging activities (see Section 2.4.2). By teaching the fundamental principles of programming, the author wished to give female students the chance to decide for themselves whether it awoke their interest. Programming should not be something intangible and mysterious, but a new opportunity to change their lives. Thus, constructionist gaming and tools that encourage learning by doing have common features: They enable working in teams, allow users to express their own ideas, and provide a visual programming language that is easy to understand and to learn. Thus, Pocket Code seemed to be a very promising tool for girls (see Section 3.2.3.2).

According to the NOLB proposal, two main goals have been set for the Austrian pilot in regard to female students:

- The gathering of background material and information about female teenagers, both from literature as well as through interviews with teenage girls in Austria, and
- To ensure a positive first experience in programming for girls with the education app Create@School.

From the literature, the assumption for NOLB was, to attract girls by (see Section 2.4.2):

- Increasing their personal attachment to programming by improving our services with appropriate example games/templates, new assets, and themed tutorials,

- Increasing their involvement by encouraging them to become active members of the Pocket Code community by providing them a safe and interesting environment to join with featured games from female users for female users, and
- Asking them to design their own games rather than just to code programs.

The intent was to discover how to develop Pocket Code in ways that specifically empower girls. Additionally, the differences in the game creation process, in game elements used, and in playing behavior have been analyzed. Aspects of self-identity and stereotyped gender categories have been taken into consideration (see Section 2.4.1.1). This was done by conducting focus group discussions and interviews to get more insights and successfully adapting the services to their needs (e.g., game jams, assets, features/functionalities, etc.).

For NOLB, the team encouraged students to be creative. Therefore, Pocket Code has been integrated in many fine arts classes, or the team created interdisciplinary courses between computer science lessons and fine arts lessons. On the one hand, students had enough time to draw their own graphics, characters, and backgrounds. To leverage these steps, the team provided a storyboard (see Appendix A.5). This storyboard consisted of four areas for the different stages of a game, or the "ceremony". On the other hand, not all girls like to draw or they think that they are not good at drawing at all. Thus, it was important that the Catrobat Media Library has been continuously expanded with themed graphics (e.g., carnival or Easter, see Figure 3.43), or graphics related to themes girls mentioned the most often, e.g., Magic, which was one result of the focus group discussion (see results chapter). Moreover, in one course, the team offered to have the graphics drawn by the TU Graz design team, which was especially appreciated by the girls.

Figure 3.43.: Themed graphics in the Catrobat Media Library.

Finally, the team come up with the idea to create a Pocket Code mascot: The Pocket Panda. At the end, our designer created a whole Pocket Code family (see Figure 3.44). They were broadly used by girls (and our whole target group) and in addition themed stickers were designed and circulated at schools.

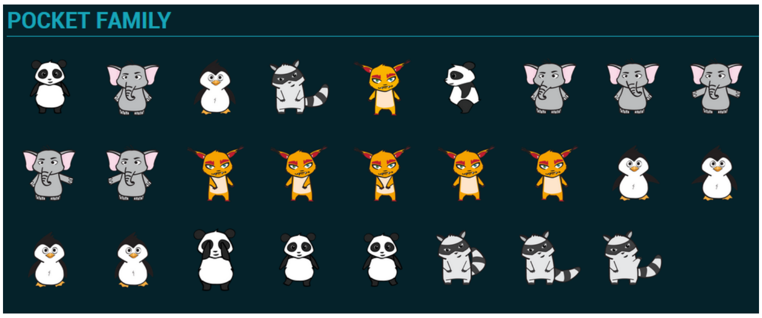


Figure 3.44.: Pocket Code family.

To summarize, this chapter first, explained the Catrobat project and its apps, and second, detailed the most important implementation of tools, services, and materials during the period of the NOLB project. Therefore, the app has been presented in more detail to provide an overview about the underlying coding concept of this app, its different screens, and parts of the app (i.e. the UI, the community sharing platform), as well as its potential to help students in learning how to code in an academic setting. Here, the option to use Pocket Code for game jams and as a tool to reinforce especially female teenagers have been presented. To provide an overview of the NOLB project and its components, the project as a whole, Austrian pilot schools, the participating teachers, and the setting of the project (cycles) have been described. Subsequently, developed frameworks, tools, and services have been explained by discussing all relevant parts of the newly developed Game Making Teaching Framework and the flavored Create@School app for schools. Finally, the last section of the chapter presented the courses performed in Austria, including their settings, used approaches, selected subjects, and topics, as well as how the team has planned to reinforce female students in particular during NOLB.

The next chapter shows all significant evaluations during the NOLB project which were either relevant for Austria, the whole consortium, or to target especially female teenagers.

Chapter 4

Evaluation of NOLB Activities and Create@School in Austria

In this Chapter, the NOLB results from the Austrian case study are evaluated in order to provide further insights when answering the three research questions proposed in the next chapter's Results. The goal for NOLB was to analyse female students' experiences, behaviors, and outcomes when using our tools and services in different courses. At the end of this chapter, an overview describing the performed game events with Pocket Code and experiences of the teachers who participated within the NOLB project is provided as well. Their intentions of using the apps after NOLB are also assessed.

4.1. NOLB Methodology

In order to evaluate the tools (Pocket Code and Create@School), frameworks, performed courses, students' programs, and all other efforts made during the NOLB project, different assessments have been conducted (see Table 4.1). The anonymity of participants was maintained at every step. For the FS, students defined their own nicknames for themselves. For the SC, teachers were given a list of usernames that had a special sequence (see Section 3.4.2). The teacher's task was to distribute the account among students in reference to their gender and to tell the NOLB team the composition of the group in which students were working together. Assessments #1 and #2 were performed during the FC together with "Evolaris"[80], a company that bridges research and economy. All other assessments were done by the NOLB team, either in all countries (Assessments #3 and #4), or in Austria only (Assessments #5-#10). These results are also part of the final NOLB delivery that summarizes the pilots per country (see Delivery 5.3 (Hughes-Roberts et al., 2017), Delivery 5.4 (Spieler and Mashkina, 2017), and Delivery 5.5 (Gaeta and Cea, 2017)). The main works of the author of this thesis were Assessments #3, #5, and #6, as well as #7 and #8, #9 and #10. Assessments #2 and #4 were the works of the whole NOLB team in Austria.

[80]Evolaris: https://www.evolaris.net

Table 4.1.: NOLB methodolgy

Evaluation of Pocket Code					
No	**Purpose**	**Data collected**	**Method**	**Number of students**	**Section**
1	Technology acceptance of Pocket Code	Quantitative and qualitative survey	User experience model	187	4.2.2
2	Usability of Pocket Code	Quantitative and qualitative survey	Descriptive content analysis	187	4.2.3
Evaluation of CreateSchool					
No	**Purpose**	**Data collected**	**Method**	**Number of students**	**Section**
3	Attractiveness of Create@School	AttrakDiff survey	Hassenzahl model assessment	350	4.3.1
4	Behavioral assessment during coding	Tracking events in Create@School	Behavioral assessment analysis	52	4.3.2
5	Usability of Create@School	Quantitative and qualitative survey	Qualitative content analysis	131	4.3.3
6	Insights on interests/values of female students	Focus group discussions	Descriptive analysis	8	4.3.4
7	Usability of predefined templates	Quantitative and qualitative survey, quiz	Descriptive analysis, qualitative content analysis	37	4.3.5
Evaluation of NOLB programs					
No	**Purpose**	**Data collected**	**Method**	**Number of programs**	**Section**
8	Game design patterns	Submitted programs (community website)	Rapid analysis (RAM), MDA	77	4.4.2
9	Learning goal achievement	Submitted programs (community website)	Significant level of learning goal achievement	271	4.4.3
Evaluation of the Pocket Code Game Jam events					
No.	**Purpose**	**Data collected**	**Method**	**Number of programs**	**Section**
10	Effectiveness Alice Game Jam	Online survey (quantitative and qualitative data)	Descriptive analysis	200	4.5.1
11	Effectiveness Galaxy Game Jam	Online survey (quantitative and qualitative data)	Descriptive analysis	462	4.5.2

In addition, for most results (n>11) the significant level has been analysed and the results were considered significant at $\alpha = 0.05$ and $\alpha = .001$. Data analysis was conducted using SPSS statistics[81] or R Project [82]. Based on the research design either a) a *t-test* (Zimmerman, 2014) was performed to compare whether two groups have different independent means, b) a *Mann-Whitney U test* (McKnight and Najab, 2010) to analyse whether the central trends of the two independent samples are different, or c) a *Chi^2-test* (Garczynski, 2014) to see if there is a significant relationship between different categories.

The following sections provide an overview about the results of the evaluation of Pocket Code (FS), Create@School (SC), the NOLB courses, and inspect students' programs in more detail.

4.2. Results of the Pocket Code App (During FS)

To evaluate Pocket Code, several quantitative and qualitative surveys have been conducted during the Feasibility Study. These surveys measured students' intentions to use Pocket Code as well as differences in subgroups, gender, and usability barriers related to Pocket Code. This section first shows the results of the quantitative survey with the help of a user experience model and continues with a usability analysis summarizing comments from the survey's open questions. In a previous study, this model was used for the first time with a smaller amount of students (Tarkus et al., 2016), and the team reused and re-evaluated it with a larger number of respondents from the NOLB pilot schools. Results of these evaluations suggest improvement for the Pocket Code app itself but also for the style of teacher training/support, preparation of tutorials and lesson content, and the backing of the whole course. These results affected the creation of the newly developed version of Pocket Code, the Create@School app. The surveys were conducted during the Feasibility Study in three phases: a pre questionnaire (T0) before starting with Pocket Code, a questionnaire directly after the last Pocket Code unit (T1), and a post questionnaire about one month after the last Pocket Code unit (T2). In each phase, a quantitative survey was administered in which students were asked different questions with the purpose of evaluating

a) Intention to use and possible barriers

b) Differences in subgroups and gender

c) Usability barriers of Pocket Code

The quantitative survey was filled out on the tablets or computers; it was analyzed with the statistic software "R" via a descriptive content analysis followed by a quantitative evaluation via Partial Least Squares (PLS) (Sanchez, 2013).

4.2.1. Study methodology.

The following sections describe the three parts of the pre-questionnaire (T0): socio-demographic factors, the technology affinity, and play behavior of the target group. The summarized preferences from the target group are very similar to those discussed in Section 2.4.2.4.

[81] SPSS statistics: https://www.ibm.com/products/spss-statistics
[82] R Project: https://www.r-project.org/

4.2.1.1. Socio-demographic evaluation.

In this survey a total of 187 students (aged 11-17) from nine different classes participated; the average age in the lower grades was 11.87 and in the upper grades was 14.51. More girls (115) than boys (72) participated. The first phase of the data collection (T0) was conducted from October to November 2015 by the whole sample (187). A total of 155 students participated in the second phase (T1), which was conducted between November 2015 and January 2016. Only 75 students completed the third phase (T2) in January 2016. The reason for this smaller quantity was that the survey had to be filled out at least one month after the last lecture, meaning by the end of January 2016 at the latest. Since three classes had their last lecture in the middle of January, they could not take part in the third survey. As a result, the evaluation was designed via two different models that measured the short and long-term experiences in reference to each student's expectations.

4.2.1.2. Technology affinity evaluation.

Technology affinity is the ability to learn about a digital culture. It encompasses a user's capacity to be responsible, competent, and confident in the usage of ICT (Bufin et al., 2015), which is important when acquiring computational thinking skills. For the study, the team was especially interested in the students' competence with mobile devices and tablets. A majority of the students (179) owned already a smartphone, and 102 of them used an android device. This is in accordance with the worldwide trend (KWUS, 2016) of mobile phone usage, (see Section 2.1.4.2). Overall, a tablet was used by 81 of the students and 137 have already used a tablet with an Android operating system. Most students use one to ten apps during one week (163) and Pocket Code was unknown to almost all students (168). Answers related to their programming experience in schools showed that almost all of them were programming novices (135). Furthermore, 52 students had already gained experience in programming at school with Scratch (31), Kodu (40), or other programs (17), e.g., C# or Visual Studio.

4.2.1.3. Play behavior evaluation.

One of the goals of this evaluation was to determine favorite game genres of the target group, their favorite playing console, and their time spent playing. Additionally, a distinction was made between girls' and boys' play behaviors. The results showed that 182 students played computer games at least sometimes, mainly on their PC, or laptop (84), or on their mobile devices (85). The most popular game genres among all students were action-adventure games (43), jump'n'run (41), adventure games (39), and skill games (39). An average of one to five hours (59) was spent with playing games per week. When comparing the frequency of gameplay between girls and boys, approximately half of the girls played computer games sometimes (49) and 8 girls played often. The greatest differences were among game genres. Most of the girls liked to play jump 'n' run (26), or skill games (23). In contrast, 61 boys said that they were playing computer games often (42), or sometimes (19). The most popular game genres among boys were sports games (29), action-adventure games (28), and skill games (28). An overview of playing behavior is presented in Figure 4.1.

Figure 4.1.: Playing behavior difference between girls and boys

4.2.2. Technology acceptance: user experience model.

The term user experience (UX) describes every aspect of the user's interaction with a product or service that influences the user's perceptions towards the product (UX Professionals' Association, 2012). This includes the layout, visual appearance, texts, sounds, and the interactions with the product. A good UX tries to coordinate these elements to allow the best experience for the user. If the UX is sufficient, it will leave a lasting impression and the chances are higher that the user will use the product again (intention of use). To measure the technology acceptance of the target group, two models with two phases have been designed to show the expectations (T0), in comparison with the short-term experience and (dis)confirmation (T1), and the long-term experience and (dis)confirmation (T2)(see Figure 4.2). Two models were used because if only one was taken over the entire period, only the sample size of T2 (n = 75) could have been used.

Figure 4.2.2 shows Model 1 and Model 2 in detail and how the measurements of influences on satisfaction (S) and intention to use (I) have been taken into account. On the one hand, the models are based on the theory of Davis (Davis, 1989), who has identified two factors that influence satisfaction (S): usefulness (U), and ease of use (E). Usefulness defines the perceived usefulness to the user themselves and ease of use means the user's expectation of how easy he or she thinks is it to use the product. On the other hand, based on Oliver's "Expectation-Confirmantion-Theory" of acceptance (Oliver, 1977), the models provide results whether these expectations of users were fulfilled or not. Therefore, both values (U, E) were compared at the time of T0 (expectations) with T1 (after the use of Pocket Code). The difference of both values between T1 and T2 are defined as the disconfirmation values (DE for ease of use, DU for usefulness) which determine satisfaction. To phrase it differently, if the expectations have been fulfilled (high amount of DE and DU), the user is more satisfied with the product, and if the expectations are not fulfilled (low amount of DE and DU) the user is unsatisfied with the product. Furthermore, the satisfaction (S) of the user has an influence on their intention to use (I). For this evaluation, the

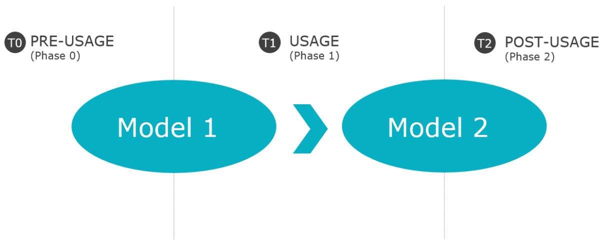


Figure 4.2.: The user experience model consists of two separate models. Model 1 is used to evaluate the short term use and Model 2 to show the impact on the long-term use.

intention to use is to be equated with the technology acceptance, which is essential for the long-term use of a product.

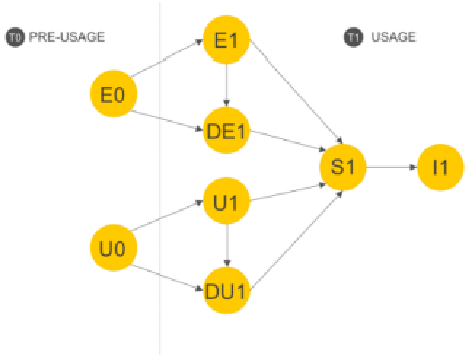


Figure 4.3.: Technology affinity Model 1 (structure level) based on Oliver's "Expectation-Confirmation-Theory" Model 1 for T0 and T1: Pre-usage and Usage. Legend: E0/E1...Perceived Ease of Use; U0/U1...Perceived Usefulness; DE1/DU1...Disconfirmation Values; S1...Satisfaction; I1...Intention to Use.

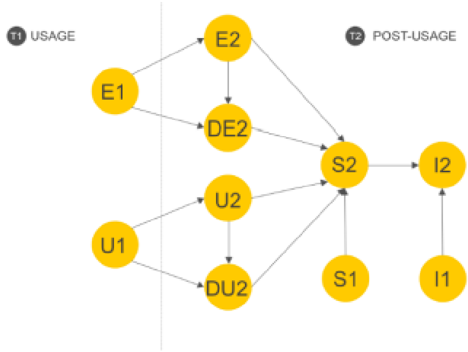


Figure 4.3.: Technology affinity Model 2 (structure level) based on Oliver's "Expectation-Confirmation-Theory" Model 2 for T1 and T2: Usage and Post-Usage. Legend: E1/E2...Perceived Ease of Use; U1/U2...Perceived Usefulness; DE2/DU2...Disconfirmation Values; S2...Satisfaction; I2...Intention to Use.

Model 2 is very similar to Model 1. In addition, in Model 2 a comparison between short-term satisfaction (S1) and long-term satisfaction (S2) as well as between short-term intention to use (I1) and long-term intention to use (I2) has been made. The collected data were analyzed with the help of the plspm-package for partial least squares (PLS) path modelling techniques (Sanchez, 2013). This technique adds values to the paths, which indicates the strength of the connections between the influence factors.

4.2.2.1. Data analysis: Model 1.

In Model 1 (see Figure 4.3), the pre-usage measurements (T0) have a significant influence on the post-usage measurements (T1). In detail, the model shows that U1 significantly influences S1 (0.71, in 1% level). In comparison, E1 influences S1 only marginally (0.15, 5% level). DU1 and DU1 have no significant influence on S1. Furthermore, S1 significantly influenced I1. R^2 shows how much of the variance is traceable. As an example for S1, 72% of the whole variance of S1 is explained by U1 and E1, which is a high indicator (see legend of Figure

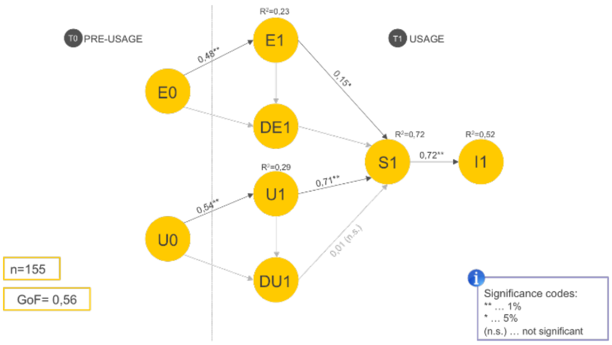
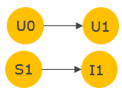


Figure 4.3.: Model 1: Estimates the parameters with the statistical software R& plspm-package. (R2 and GoF) should be high: (R2, GoF) < 0.30 (low)-0.30< (R2, GoF) < 0.60 (moderate)-(R2, GoF) > 0.60 (high).

To learn about differences among subgroups, a permutation testing (Nichols and Holmes, 2001) was used to compare two groups of people. Against expectations, the results showed that there are no significant differences between the subgroups in T0 and T1. Tested subgroups comprise of the following: girls and boys, younger and older students (lower grade/upper grade), students with or without programming experience (or prior knowledge of Pocket Code), students who own or do not own a tablet, or students with different playing behaviors. A significant difference is only visible between the various schools (see Figure 4.4). This outcome shows that for students at Borg Birkfeld and Akademisches Gymnasium the expectation of U0 had a significantly higher influence on U1 (0.80 and 0.50) than at the other schools. To phrase it differently, these students had very positive feelings about the app Pocket Code before actually using it, which further significantly influences S1 and I1.

The second row of the table in Figure 67 shows that for those classes who had bad expectations and experiences in U0 and U1 of Pocket Code, the S1 significantly influences I1 (0.81).



Global (n=155, T1)	GIBS (n=86)	BORG (n=27)	Akadem. (n=42)
0.54	0.27	0.82	0.50
0.72	0.81	0.62	0.40

Figure 4.4.: Differences between the three pilot schools (3 group comparisons were necessary).

4.2.2.2. Data analysis: Model 2.

In Model 2 (see Figure 4.5), the pre-measurements (T1) again have a significant influence on the post-usage measurements (T2), and an especially significant impact is seen on U1 for U2 (0.83) and the associated DU (0.29). E1 is statistically important only at the beginning (0.77) but has no influence in the long-term use (0.14) for S2 and I2. This means that for long-term use, U is the most important and for novices E is essential as well. The most important takeaway here is the impact of S2 on I2, and the interpretation that if a user had the intention to use it at the time of T1 he will also use it later on (T2). In other words, the short-term intention to use influences the rate of long-term acceptance. The GoF for this model is in comparison to Model 1 higher and the variant (R2) has a sufficient value as well.

To summarize, differences within the subgroups have been evaluated (see Figure 4.6). On the one hand, no significant differences between T1 and T2 are visible between the subgroups of younger and older students (lower grade, upper grade), students with or without prior knowledge of Pocket Code, or students that own or do not own a tablet. On the other hand, the results indicated a significant difference between the observed subgroups of girls and boys, students with or without programming experience, and students' playing behavior. In detail, for male students S2 has a more significant influence on I2 (0.79) than it has for female students. Figure 4.6 demonstrates the differences between Model 1 and Model 2 in the path coefficients by comparing the user's gender. Additionally, for students without programming experience the short-term U1 plays a significantly higher role (0.83) for the long-term U2 than for students with programming experience. To put it succinctly, if users do not know what programming is about, Pocket Code seems to be a suitable tool to learn about first steps of programming. Both results go along with the third row of Figure 4.6. This indicated that the more students are used to playing computer games or play games on a regularly basis, the more important S2 is for I2 (0.56 and 0.77). Compared to students who play seldom computer games S1 had only a minor influence on the further usage of Pocket Code (0.13).

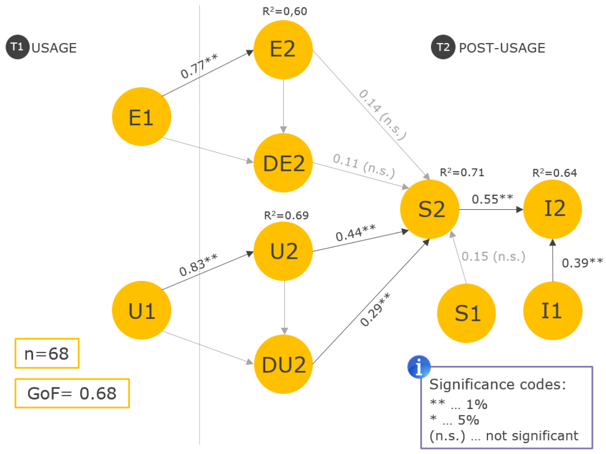


Figure 4.5.: Model 2: Estimates the parameters with the statistical software R & plspm-package. (R2 and GoF) should be high: (R2, GoF) < 0.30 (low)-0.30< (R2, GoF) < 0.60 (moderate)-(R2, GoF) > 0.60 (high).

Differences between

- Girls and boys:

Path	Global (n=68**)	Girls (n=46)	Boys (n=22)
S2 → I2	0.55	0.40	0.79

- Pupils with or without programming experience:

Path	Global (n=68)	Experience (n=11*)	No Experience (n=57)
U1 → U2	0.79	0.59	0.83

- Pupils that play computer games never/sometimes/often

Path	Global (n=68)	Never (n=16*)	Sometimes (n=39)	Often (n=13*)
DU2 → S2	0.29	0.18	0.07	0.70
S2 → I2	0.55	0.13	0.56	0.77

Figure 4.6.: Differences among the groups (permutation testing).

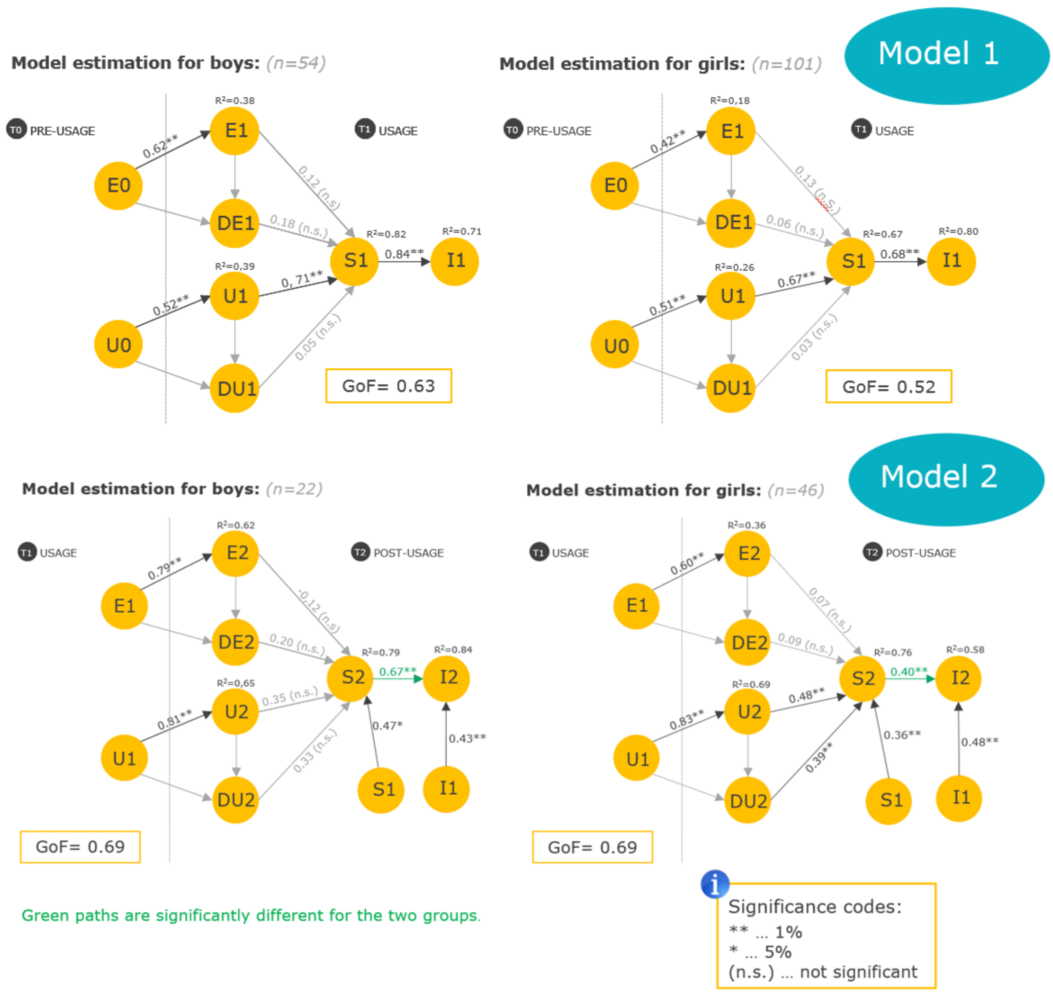


Figure 4.7.: Comparing groups: male/female students; permutation test for differences in path coefficients.

4.2.2.3. Discussion: user experience model.

During the Feasibility Study, the team not only collected the data of the surveys but also took notes, made pictures, and interviewed students while working with Pocket Code. Observations showed that overall all of the students enjoyed using Pocket Code in class to work in groups, share ideas, and to develop games together. However, it could be noticed that not all students participated equally in crafting and coding. Many students adopted roles they were comfortable with. Two different divisions within the teams could be observed. First, one student was programming while the other one was drawing the characters, objects, and backgrounds. This happens mostly in female teams. Second, the team split their work so that everyone was programming one level of the games, which was later merged into one big game. Thus, the challenge for the workshop facilitator was to also motivate these students that spent most of time drawing the artwork to code some parts on their own. By comparing the results of the survey, the differences between the schools could be clearly explained. Students at Borg Birkfeld and Akademisches Gymnasium were given more instructions and help (A1) than schools that had a more constructionist approach (A2 and A3). The team observed that students who were told to explore Pocket Code on their own in a short time span felt more stressed and asked more questions than students who got a framework at the beginning. Further, both schools appreciated this new playable programming access with Pocket Code more. As a result, the students were highly motivated, which can be seen in their perceived U1 and U2. At Borg Birkfeld, students in course number 20 had already done some programming, e.g. with C#, but had some bad prior experiences with programming and a strict the teacher. For them, Pocket Code was a welcome diversion from the usual computer science lessons. The importance of a positive first experience in coding (especially for girls) has been already described in Section 2.4.2 and is also visible in Model 2. In addition, students at Borg Birkfeld had fewer Pocket Code units (two double units). Thus, they may not have detected problems with the app, which may occur during long-term use. Furthermore, students at Akademisches Gymnasium had very motivated teachers and were allowed to create and draw their characters on their own. Their goal was to present the game on the open house day of their school.

At GIBS, the students seemed to be very frustrated after the last Pocket Code unit was performed. According to the on-site observations, the reasons for this was threefold. First, some students had already programmed with Scratch, and some functionalities from Scratch were missing in Pocket Code. Second, the teachers demanded a lot, and assessed every step of the project (storyboard, presentation, and program). The students felt stressed by this pressure from their teachers as well. Third, some students had very broad ideas at the beginning. When they became aware of the complexity of their ideas and the workload that was needed to achieve these goals, they then limited their initial ideas to what they were able to implement in that short amount of time. Although they had many programming units (10 double units), their programs did not implement their initial ideas and they felt disappointed as a result. All of these factors influenced their S1 and I1 and are also present in Model 2. In Model 2, the difference between male and female students is interesting to observe. Section 4.2.1.3 shows and the literature argues (see Section 2.4.1.5) that male students have a higher technology affinity and play games more often than female students. Thus, they have more experience when comparing their creations to similar products and expect a higher level of satisfaction than female students expect.

On-site observations indicated some slight differences in how the different genders approached their tasks. Male students were more concerned with the gameplay itself; the game should be fun, fast, and challenging. Girls mostly started with only a rough idea, and followed that with designing and drawing their gaming worlds with a lot of care. As a consequence, female students often had less time to code and get the game to a playable state (i.e. they felt more stressed). The developed technology acceptance model shows dynamic changes of acceptance over the period of the Feasibility Study, thus contributes to the understanding of the user's intention of long-term use. The Pocket Code units (short-term use) satisfied students the most if they were already motivated at the beginning (i.e., they saw the usefulness of the service), e.g., through a playful approach, useful introduction, or a motivated teacher/role model. In addition, students who used a framework or got more help at the beginning (i.e. ease of use) were more satisfied and intended to use Pocket Code further. For the long-term use of the app, students who started without any programming experience found Pocket Code more useful; the effects of having these "aha" moments profoundly influenced their rate of satisfaction.

Block based coding languages are indeed to help novice programmers (see Section 2.1.4.1). Pocket Code seems to be a suitable tool for (female) students who want to learn how to code and may significantly contribute to closing the divide and participation gap in digital culture. The on-site observations also showed the significant promise of getting the youth involved in the world of coding with Pocket Code. The tool helps to bring their ideas to life and create something that is meaningful for them. However, concerning the workload of teams, it seems to be more efficient if every student begins by working on their own level, thus feeling ownership about the game. Nevertheless, the entire group should be working on the same bigger game. At this time, the merge-functionality was still beta and did not work sufficiently.

For the next cycle the team noticed that it was necessary to discuss the storyboards or the game ideas in more detail together with students, in order to know what was possible in the available amount of units. According to the amount of units, the team concluded that the appropriate amount depended on the approach. With A1 two double units for the main learning (i.e., programming) are already sufficient whereas for the students' personal games two double units are the minimum and four the maximum. Although Pocket Code is a promising tool that encourages novice programmers to become game designers and creators, the app should also fit to the needs of students who have already gained some experience in programming or those who often play in their free time. Thus, this survey also suggested feature completeness in Pocket Code to program games that are more advanced. This could be seen as well in the results of the usability evaluation, which is part of the next section.

4.2.3. Usability: descriptive content analysis.

As has already been discussed in Section 2.1.4.2, mobile learning is seen as a way to offer flexibility in learning and presents a lot of educational advantages, i.e., the learning process becomes more independent of the spaces previously determined for their use (OECD, 2004). Many concepts in programming are difficult to understand for younger students. Therefore, it is important that the learning tool supports students well in their task and does what it is supposed to do. This is summarized with the term usability (Bhattacherjee and Premkumar, 2004). For a product to be seen as useful or user friendly, it must help to reach goals effectively, efficiently, and satisfactorily. This can be achieved by simplicity, design, ease of use (see previous section), and consistency in navigation, as well as structural, metamorphic, or standardized iconic design. This is important to meet a range of cognitive learning styles, to maximize the learning experience, and to increase the learning curve indirectly (Hubwieser, 2007). This section evaluates the usability of Pocket Code and therefore describes the barriers of Pocket Code as well as its limitations and advantages as summarized through the surveys of the target group. Additionally, several open questions have been asked in T1 and T2. In addition, T2 included some questions concerning the future use (i.e., in spare time) of Pocket Code, which is part of Section 4.2.3.2.

4.2.3.1. Usability data: Pocket Code.

This section presents the usability evaluation of Pocket Code by answering like/dislike questions and offering recommendations.

Q: What did you like best about Pocket Code? 147 mentions (T1) and 70 mentions (T2)

Figure 4.8 presents the students' answers. What students liked the most was the aspect of creating something and learning something new.

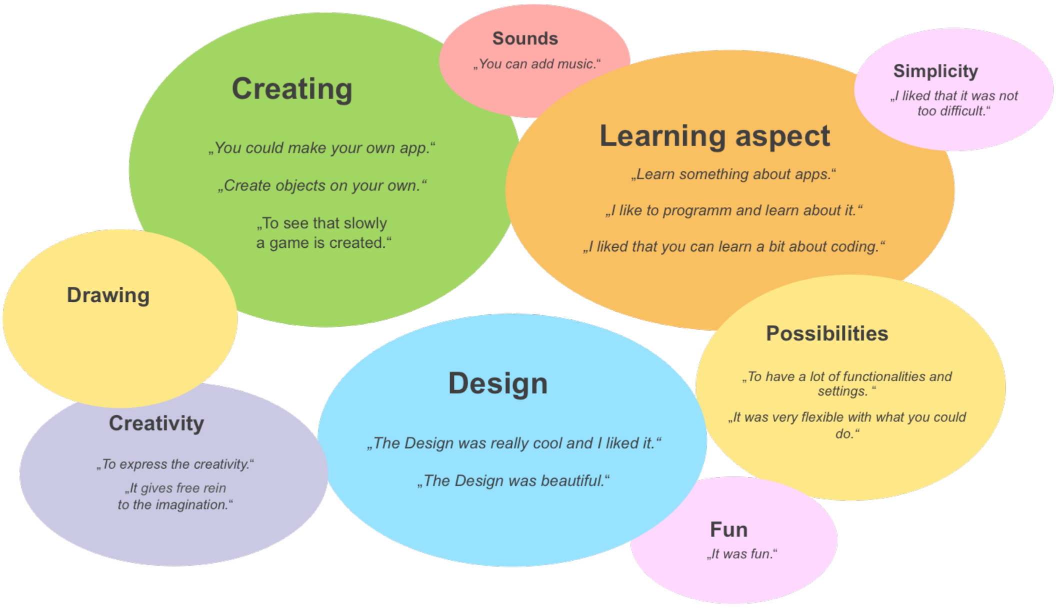


Figure 4.8.: Answers to the question: "What did you like best about Pocket Code?" The size of the bubbles correlates with the frequency of the mentions.

What did you like least about Pocket Code? 136 mentions (T1), 60 mentions (T2).

Many students mentioned that the app often crashes and that Pocket Code was a little bit too complicated (especially at the beginning).

What is the most important thing that should be changed in Pocket Code? 118 mentions (T1) & 58 mentions (T2)

Students suggested that the design should be simpler and clearer, and functionalities should be improved as well as the general performance of the app. Answers are visualized in Figure 4.9, including recommendations.

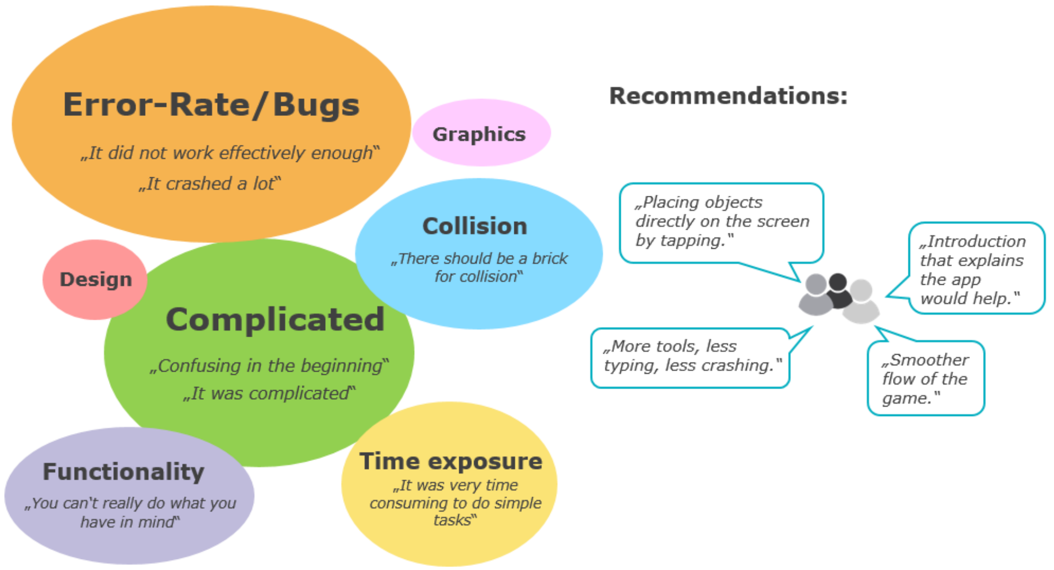


Figure 4.9.: Answers to the question: "What did you like least about Pocket Code?" The size of the bubbles correlates with the frequency of the mentions.

Additionally, questions according to the formula editor have been asked. The formula editor has been described in more detail in Section 3.2.2. It is needed, e.g., to access the device's sensors, to manipulate the object's attributes, to insert different formulas, and to use variables/lists within conditions. Prior observations showed that these functionalities are especially difficult to understand (especially for younger students), hence the survey included questions regarding the formula editor in T1.

How do you like the formula editor? n=153

Students evaluated the formula editor with an average of 2.8 (see Figure 4.10).

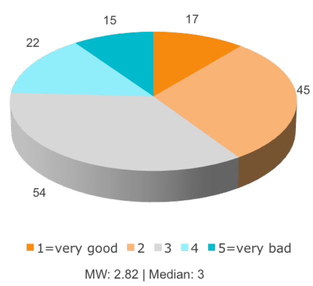


Figure 4.10.: Evaluation formula editor-average of 2.82 and median of 3.

What is the most important thing that should be changed in the formula editor of Pocket Code? n=104

Suggested usability improvements for the formula editor included to make it more simple (i.e. regarding clearness and terminology), add more descriptions and help options, improve scalability of the buttons (to avoid misclicks), and make the use of variables easier (e.g. with simple bricks).

4.2.3.2. Further usage: Pocket Code.

The team expected a higher number of students who intend to use the Pocket Code app further after the Feasibility Study. Pocket Code has been used by 8 students at least sometimes in their leisure time for an average time of 1.2 hours. As a school project, Pocket Code was used by 44 students. Their intent to use Pocket Code was either because they wanted to program a game, they took part in the Alice Game Jam event (see Section 4.5.1), or because they just tried it out for fun. However, most of them, mentioned that they had no interest (20), or time (16) to use Pocket Code further in their leisure time. Other answers include that they found it not useful (12), too difficult (8) or not as fun (5). Since Pocket Code is only available for Android, some students (4) mentioned that they only have an iPhone or iPad, and some noted problems when downloading the app (3). In addition, the survey had two more open questions to answer:

What would enhance your motivation to continue using Pocket Code? Answers include: if the app would be easier to use, would become more user-friendly, or would offer more assets for creating games.

What would decrease your motivation to continue using Pocket Code? Answers include: no improvements in error-rate/bugs, if it became more difficult, advertisement within the app, or a lower quality of games.

While comparing future usage between girls and boys, it seemed that boys used Pocket Code more often (6) and longer in their leisure time (average hours 2.1) than girls (2, average hours 0.3).

4.2.3.3. Discussion: usability evaluation.

Although the Feasibility Study and the interaction with students was overall very positive, it also determined need for improvement. Enhancements were recommended not only regarding the backing of the course and the teacher training/support, but also of the Pocket Code application itself. Pocket Code may have helped its users to learn basic programming steps but it was difficult to program advanced or large games. The app needed improvement in terms of the usability by decreasing both the complexity and the time needed to develop games.

To increase scalability of Pocket Code beyond computer science classes, the following suggested improvements were considered and they influenced the development of Create@School the most:

- Providing hints for beginners
- Adding improvements to the Formula Editor (FE)
    - — Intro which explains the important functionalities
    - — Increase size of the FE buttons according to device screen (especially for tablets)
    - — Add subcategories within the sections of the FE
    - — Possibility to rename strings, variables, and lists
    - — Modify the way of deleting
- A more accurate and easy to use detection between objects (collision detection)
    - — Physical collision detection (using Box 2D)
    - — Collision property within the FE (e.g., if object touches object, edge, finger)
- Improvement of functionalities, e.g., the merge functionality

- New functionalities:
  - — Backpack (to copy and paste scripts, looks, sounds and whole objects)
  - — Single or multi-touch option
  - — New loops/conditions: repeat until, wait until, etc.
  - — Drag and drop of elements within the UI
- A range of new bricks:
  - — Ask brick (to ask questions during the game) either via a input box or speech recognition
  - — Bubble bricks (for say and think bubbles, for storytelling)
  - — Clone bricks (create clones of the object)
  - — Pen/stamp bricks (to let the object draw patterns/a line)
  - — Speak/play sound and wait brick (wait did not exists before)
  - — Stop script brick (to stop this scripts, all other scripts, or all scripts)
  - — Bricks to detect background switch (new hat brick, for switching between backgrounds)
  - — Rotation style brick (e.g., that the object does not flip if it bounce off the edge)
  - — Go to brick (go to another object, finger position)
  - — Previous look brick (to switch also to the previous look)
- New formula editor properties
  - — Look/background number/name (e.g., to ask for a specific look name/number)
  - — Values for longitude, latitude, location accuracy and altitude (GPS)
  - — Current time (ask for the current year, month, day, etc.)
- Usability improvements:
  - — Separate events from other control bricks (like in Scratch)
  - — Opportunity to comment out bricks or whole scripts

Based on these results, the NOLB team developed Create@School and reflected strategies on game design to create a more productive experience for learners. In addition, the development team focussed on code refactoring/quality and test coverage to improve the performance and stability of the app as a whole.

4.3. Results of the Create@School App (during SC)

During the Second Cycle of the NOLB project, Create@School was evaluated several times. Results were collected through the following evaluations:

- Usability, satisfaction, and attractiveness evaluation by using the Hassenzahl model assessment
- Behavioral assessment evaluation through tracked events within the app

- UX surveys with open and closed questions
- Usability test with two pre-coded templates (physical simulation and RPG adventure template)

Since the Create@School app and provided services should serve to support female students, the focus of all evaluations was to identify differences in gender. The attractiveness of the app to female students, their programming behavior, and usability experiences with either Create@School or the pre-defined templates were all assessed in order to get insights about their personal interests and values.

4.3.1. Attractiveness of Create@School.

During the Second Cycle, an AttrakDiff survey (User Interface Design GmbH, 2018) was conducted after the last Create@School class. A research which conducted two studies (Schnurr et al., 2017) with different contexts and different test subjects came to the result that consumers perceive unfamiliar technology as more attractive when it is placed in an attractive context. Moreover, they rate it as higher quality, which translate into higher intentions to purchase it. Attraction is also a component of the requirements catalog for mlearning environments from (Freitas et al., 2013) (see Section 2.1.4.2). Mobile devices should increase one's motivation in using and learning through the environment itself. For the user, this new modality of learning should foster curiosity, contributing to better understanding information that has been taught. Attractiveness surveys in general ask users to assign different attributes to a product or service and give additional questions about the situation in which the product or service is used. Collected data from the AttrakDiff survey is evaluated through a Hassenzahl model (Burmester and Koller, 2008). This model evaluates:

- the usability and utility of the technologies perceived by users,
- the satisfaction of users that have used the technologies, and
- the attractiveness of the technologies.

This section first explains the Hassenzahl model in general and second presents the outcome of the model. Therefore, the type of the survey has been set to "Comparison A and B", where A collected the answers from female students, and B collected answers from male students. To conclude this section the results of the Hassenzahl model are discussed.

4.3.1.1. The Hassenzahl model.

The Hassenzahl model gives an insight on how usable the app was for its users by highlighting the values of hedonic and pragmatic attributes assigned to the product by users (Law et al., 2017). The AttrakDiff questionnaire is a survey based on the Hassenzahl model, which builds a quantitative scale to measure both hedonic and pragmatic qualities. It is a psychometric tool based on a semantic differential. The relationship with technology affects people's emotions and quality of life. The Hassenzahl model is designed for measuring both the user's experience and satisfaction and tries to measure their emotional responses. Based on the Hassenzahl model, the qualities of physical products, websites, software, and other digital media can be classified into two distinct groups:

Pragmatic qualities: These qualities are related to practicality and functionality. An example for pragmatic qualities is usefulness and usability. Examples of attributes that are typically assigned to software in general are supportive, useful, clear, and controllable. The purpose of the software should be clear and it should be understandable for the user.

Hedonic qualities: These qualities reflect the psychological needs and emotional experience of the user. Within the Hassenzahl model hedonic qualities are divided into two categories:

- *Stimulation (HQ-S):* The user wants to be stimulated in order to enjoy the experience with the software or product. Rarely used functions can stimulate the user and satisfy the human urge for personal development and increased skills.
- *Identification (HQ-I):* Users have a need for expressing themselves through objects and they like to control how others perceive them. All people have a desire to communicate their identity to others. This is done through the things people own and use. These things help them to express themselves, define who they are, and show what they care about. This is why people enjoy using personalization on sites such as Twitter[83]. Changing certain items, e.g. the background wallpaper and header image, allows people to express themselves.

The right balance between pragmatic and hedonic quality is important to give the user a positive experience and satisfaction while working with the application. Usability specialists commonly use the semantic differential (Strungúa, 2014) in order to overcome the poverty of subjective data that would otherwise be obtained from tests or an open question survey.

4.3.1.2. Hassenzahl model assessment.

The Hassenzahl evaluation is done in the form of a survey consisting of the 28 pairs of terms with polar opposite meanings, for instance, "motivating" or "discouraging". The user has to cross one of the six circles between each pair according to user's personal view of the product (Burmester and Koller, 2008), (see Figure 4.11) as an example. The whole AttrakDiff survey is part of the Appendix (see Appendix A.3).

Figure 4.11.: Example word pairs of the AttrakDiff survey.

For this evaluation also the students from the Spain pilot filled out the survey. In total, 152 girls and 198 boys participated in the survey used to construct this model. In UK the used another survey because the UK team considered the AttrakDiff survey as too difficult for their younger target group

[83]Twitter: https://twitter.com

and students with learning disabilities. This evaluation is part of the NOLB Delivery 5.4 (Spieler and Mashkina, 2017).

As can be seen in Figure 4.12, Create@School received neutral evaluations for pragmatic and hedonic perspectives from both genders. The small confidence intervals signify that the results are reliable and not coincidental. Male students (cyan) rated the app as slightly more attractive than female students (orange) did.

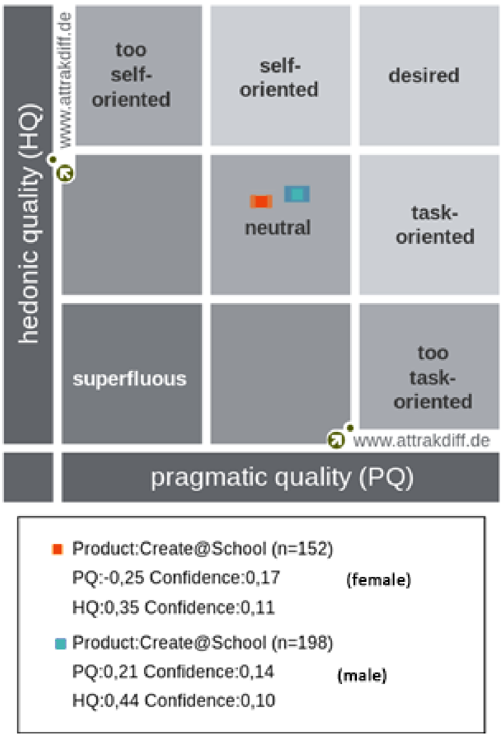


Figure 4.12.: Overall evaluation of hedonic and pragmatic qualities.

The diagram of the average values can be seen in the Figure 4.13. The average value of the pragmatic quality (PQ) evaluation has the largest absolute distance of 0.46 between female (orange line) and male students (cyan line). The average value of the PQ rated by the female participants has the value -0.25 and thereby it is below 0. The identity (HQ-I) and attractiveness (ATT) average value is slightly larger for the male participants with the difference of 0.13 and 0.18 respectively. The stimulation (HQ-S) has approximately the same average values with 0.53 from female and 0.57 from male students. The average values of the hedonic qualities do not reach maximum values in any of the aspects.

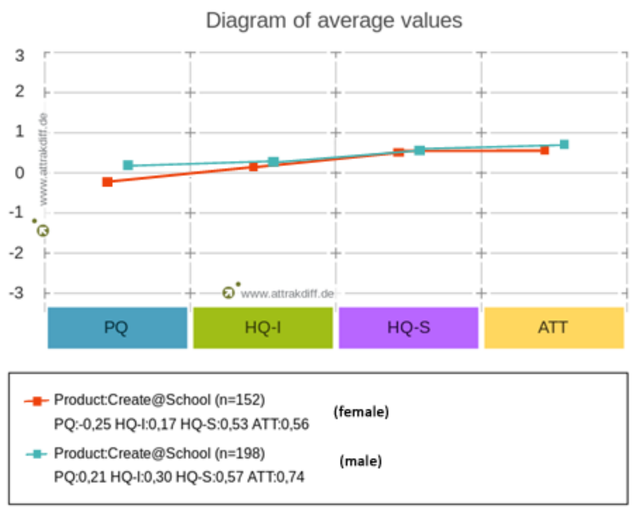


Figure 4.13.: Diagram of the average values for pragmatic qualities (PQ), hedonic qualities — identity (HQ-I), hedonic quality stimulation (HQ-S), and attractiveness (ATT).

The average diagram shows that:

- almost all qualities achieve a positive evaluation in both genders,
- the largest distance between the genders is seen in pragmatic quality,
- female students' evaluation of pragmatic qualities is less than zero,
- for identity (HQ-I) male students rate Create@School slightly better than female students,
- for stimulation (HQ-S) female and male students' rates are almost the same,
- for attractiveness (ATT) male students rated Create@School slightly better, and
- no quality reaches the maximum rating.

The diagram of Figure 4.14 shows the details of the semantic differential of the model, the adjectives of the survey, and the values that have been contributing to each quality. Overall, none of Create@School's qualities reaches the maximum value. The negative feedback is mostly given by female students, except the "cheap-premium" pair has been rated more negatively by male students.

Female participants considered the app more technical than human, as well as more complicated than simple and confusing rather than clearly structured. Male participants also rated the app more technical than human, but at the same time found it more practical than impractical, more straightforward than cumbersome and more structured than confusing. Both groups had a neutral opinion on the predictability of the app.

The identity (HQ-I) as well as the simulation (HW-S) evaluation of the app are mostly consistent between the two groups. Both groups found the app cheap rather than premium and professional rather than unprofessional, as well as more stylish than tacky and presentable than unpresentable. The participants had a neutral opinion about the alienating-integrating behavior of the app. Male students found Create@School more connective than female students did. Girls said that the app separates them from people rather than brings them closer to people. Both female and male students considered the app as more inventive, creative, innovative, and novel. Boys rated the app as more bold and captivating than girls did. Male participants assessed Create@School as slightly undemanding, rather than challenging, whereas the female participants saw the app as challenging.

Both student groups rated the attractiveness (ATT) of Create@School positively. The boys rated the app as more likeable, inviting, appealing, and motivating than girls did. Female and male participants found the app rather more pleasant than unpleasant, attractive than ugly, and good than bad.

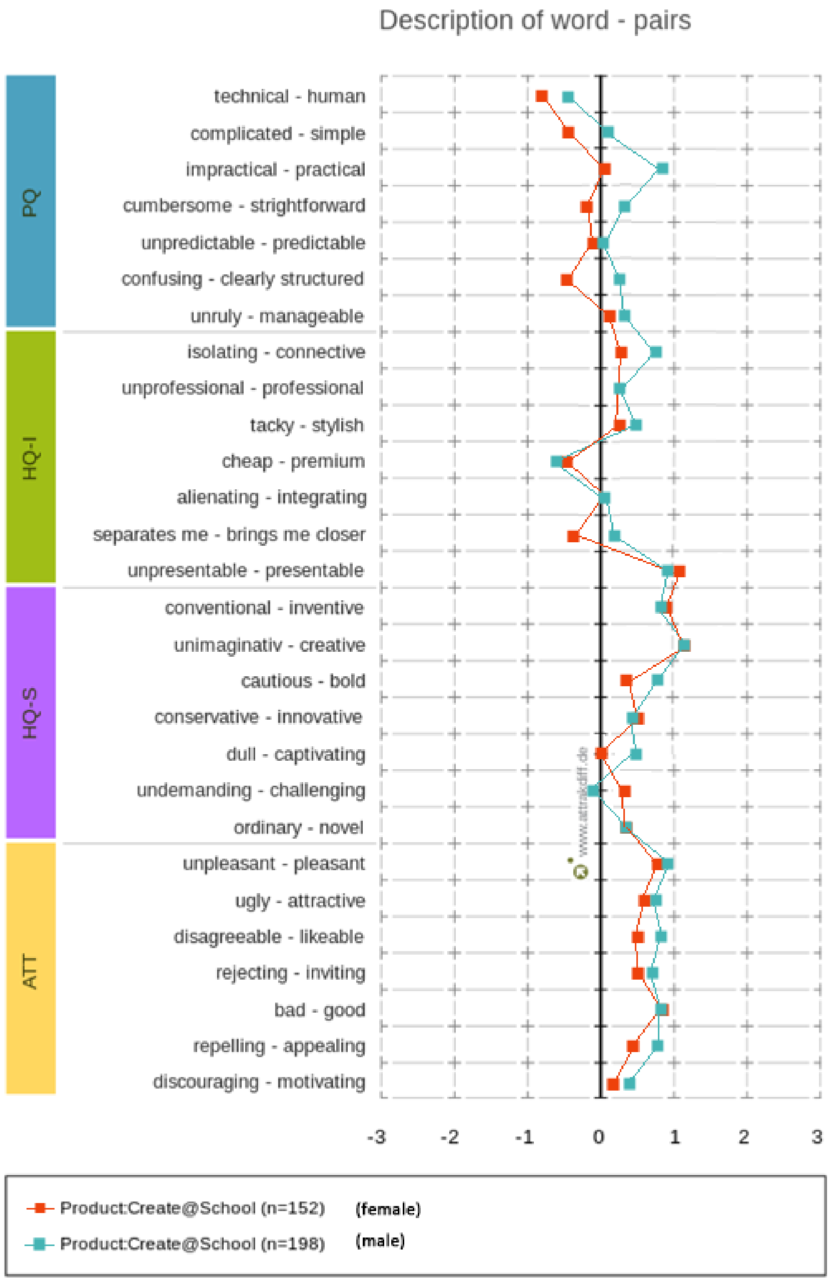


Figure 4.14.: Attractiv survey: description of word-pairs.

4.3.1.3. Discussion: Hassenzahl model.

Create@School received very similar evaluations from both genders. In general, all students accepted the app and ranked it positively even though it has not yet achieved the desired acceptance level. The positive values recorded and the good balance between pragmatic and hedonic qualities shows a degree of satisfaction while working with the application.

These results of PQ are not surprising and correlate with the literature in Section 2.4.2.5 which states that there are differences between the perceptions of tools between the genders. On-side observations showed that male students were more interested in testing the various functions of the app. They also programmed at home (even if they were not asked to do so), and were more interested in creating something on their own than just finishing the given task. For the female students it seemed to be more a means to an end, which meant they were more concentrated on creating the game which the team or the teacher asked for and less interested in tinkering around with the app. This correlates with the findings in Section 2.4.1.5 that the ability of feature discovery (i.e., to try out new features) differs between female and male students. The especially positive ratings for creativity and presentability show that the participants enjoyed their experience with Create@School in general and that the app allows users to express themselves. Regarding the negative rating for "it separates people" rather than "brings them closer", further research needs to be done if this somehow is an indicator and influences female students' sense of belonging or self-efficacy negatively. The question is if students who work individually on games but with a similar task either feel more engaged or disengaged with the whole group.

To conclude, the Create@School app was perceived as useful and usable, attractive and stimulating, and as being a good foundation product for the creation of a new innovative tool for education which promotes the social inclusion of students. Therefore, the next section provides practical insights into the programming behavior of students, thus showing students' actual interests, skills, and creativity.

4.3.2. Behavioral assessment: use of Create@School.

Behavioral assessment (Feindler and Liebman, 2015) is the approach of understanding and changing behavior by identifying the context in which it occurs (the situations or stimuli that either precede it or follow from it). In other words, it is measuring a person's actual behavior (The Center for Emotional Health, 2016). Once the behavior is defined and measured, different factors were defined that may reinforce and maintain this behavior. Results involve the recording of the frequency of various behaviors, understanding the interactions between situations, and finally, actions for change in behavior.

Cognitive and behavioral measurement in NOLB has been handled through qualitative (observation) and quantitative data (automatic collection data from usage of Create@School), (see Section 3.4.2.4). Through statistical analysis of collected data evidence, feedback is being provided to improve the constructivist-oriented learning and teaching model. This is done by evaluating several users' cognitive/didactical and behavioral performances, such as engagement, persistence, and Create@School usage. The next section shows the results of the behavioral assessment of three classes from the Austrian pilot schools. The whole evaluation is part of the NOLB Delivery 5.4 (Spieler and Mashkina, 2017).

4.3.2.1. Behavioral assessment analysis.

The analytics data tracking was part of the Create@School app. At the beginning in October 2016 this data were used mostly for testing purposes and thus the tracking has consciously improved (e.g., addition of missing events). In March 2017, the project id which is added by students was tracked as well (provided by the teacher, same as in the PMD). This identifier is unique per class project and allows researchers to determine which students belong to the same class. This was important because all events were not only analysed per each student but also in reference to the class he or she belongs to. Therefore, the average of each event and also the maximum of events used by a student in each class could be measured. The PMD was ready for use in schools in April 2017. Thus, only three classes in Austria could be evaluated because the other classes already finished their coding lessons. The behavior measurements are available for the courses 11, 17 and 18 (two courses but the same students), and course 19 (number of students in total n=52). The composition of each parameter as well as the grades is already described within Section 3.4.2.6.

In the following sections the results of the collected data are presented; first, for all classes in general and second, with a focus on gender. For the analysis only the excel dashboard will be used because they allow more customization.

Behavioral analysis in general

First, the behavioral analysis is provided for all classes and next the classes are compared. Figure 4.15 provides an overview of the behavioral analysis of all three classes together. The parameter "Usage of Create@School" and "Confidence" (in green) reaches almost the maximum value (>3.5), whereas "Creativity" and "Positive Affect" (in orange) reaches the lowest value of 2.48 (but still above average). This means that students created and coded a lot (.i.e., used the app), the games met the learning goal, and at the same time they had the chance to create their own assets via Pocket Paint, etc.

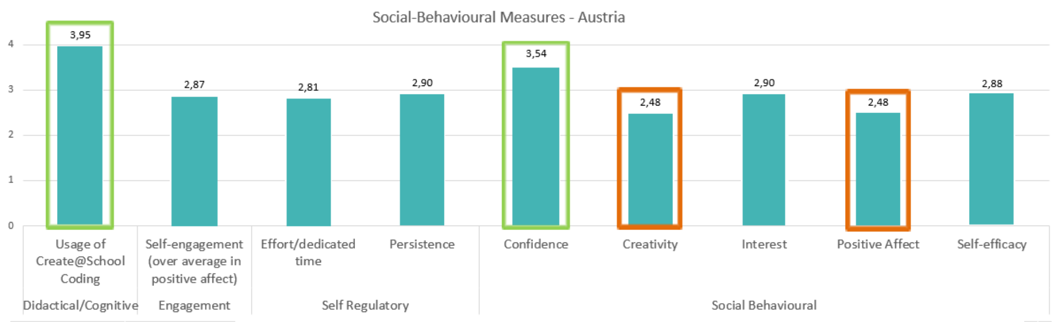


Figure 4.15.: Behavioral analysis results of all three classes.

The differences between the classes can be mostly be sorted by the different approaches used. Figures 4.16, 4.17, and 4.18 showed the results per class.

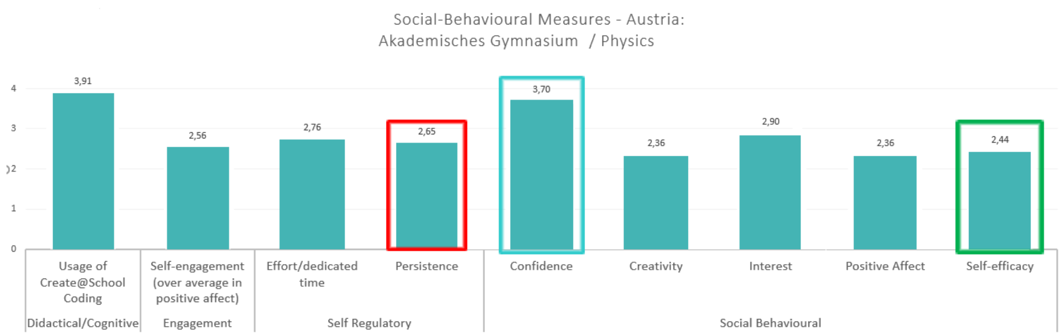


Figure 4.16.: Social-behavior measures: Akademisches Gymnasium Grade 7/physics.

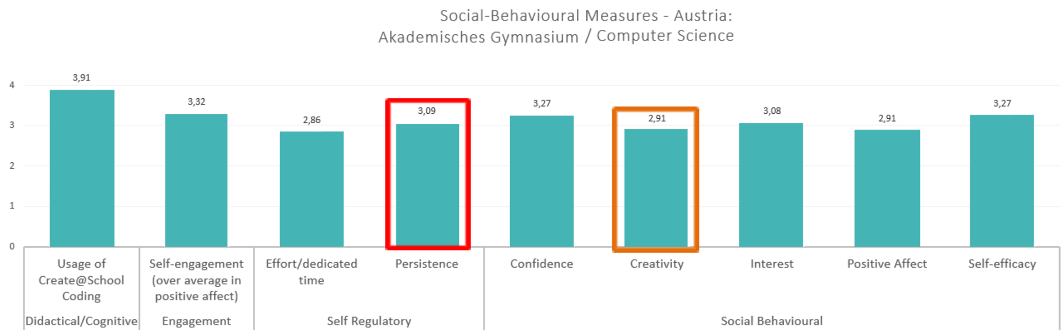


Figure 4.17.: Social-behavior measures: Akademisches Gymnasium Grade 12/computer science.

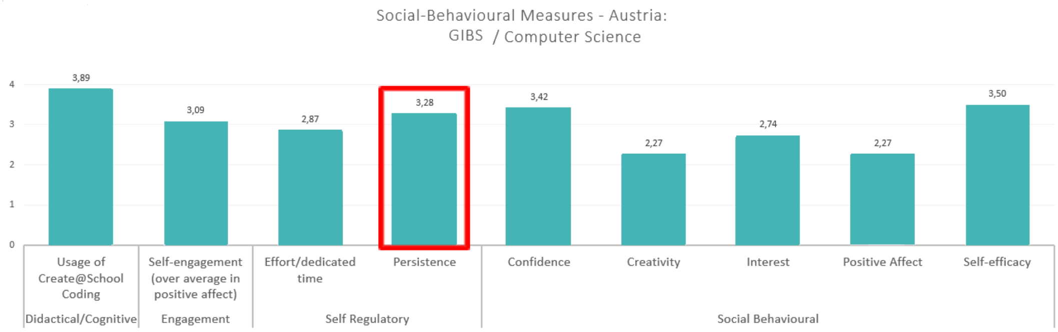


Figure 4.18.: Social-behavior measures: GIBS Grade 9/computer science.

The parameter “Self-Efficacy” (in green) is significantly lower (2.44) of students in Figure 4.16 than in the other two classes (Figure 4.17: 3.27 and Figure 4.18: 3.50). The reason is that students from Figure 4.16 only had 3 coding units at 50 minutes, whereas students from Figure 4.17 had 8 units (and worked on their games at home as well), and students from Figure 4.18 had 10 units (one student worked at home). The parameter “Usage of Create@School” is significantly high in all three classes (3.90). This means students had a high interaction with the app in all classes. The parameter “Creativity” (in orange) is slightly higher in Figure 4.17 (2.91) than in the other two classes (Figure 4.16: 2.36 and Figure 4.18: 2.27). This class used, the Quiz template in the second course but first created their games from scratch. Thus, they created more assets on their own than the class in Figure 4.18, which used the pre-coded templates. The class in Figure 4.16 used mostly camera assets for their games. The value of the “Confidence” parameter is slightly higher in Figure 4.16. In this class, almost all reached the predefined learning goal set by the teacher.

A notable difference could be seen in the “Persistence” rate between the three classes (in red). Therefore, Figure 4.19 presents this parameter in detail to see the different values.

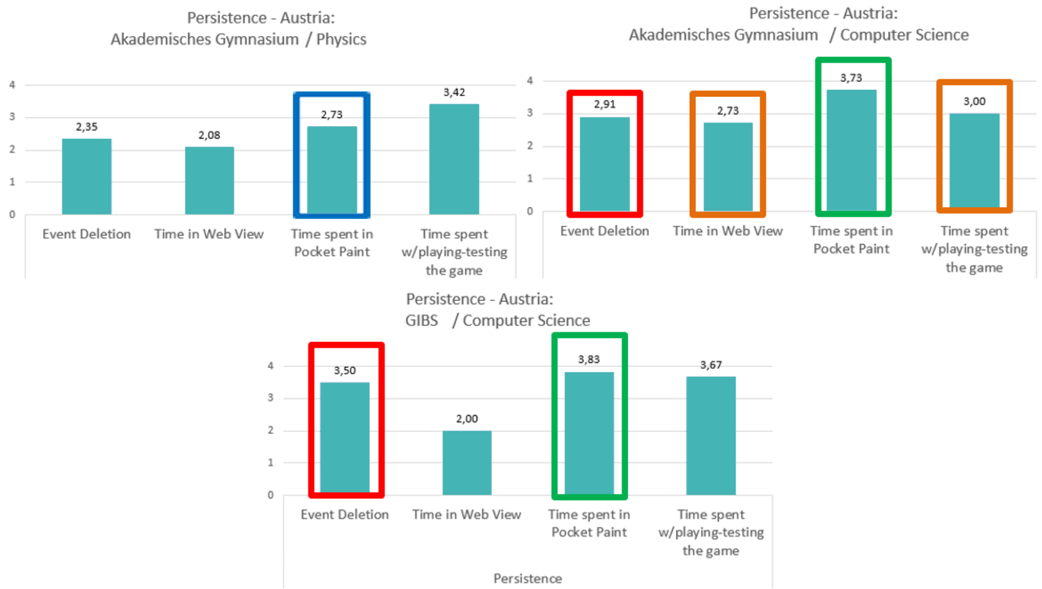


Figure 4.19.: Persistence in more detail. a.) Akademisches Gymnasium/Grade 7, b.) Akademisches Gymnasium/Grade 12, and c.) GIBS/Grade 9.

Students in Figure 4.19 c) showed a high value in “Event Deletion” (3.50, see bars bordered in red). This class used a template (adventure RPG). Therefore, they deleted scenes, objects, and looks that were likely not those which were needed for their games (3.50). In addition, students in Figure 4.19 b) first used a template (2.91) whereas the value of students in Figure 4.19 a) is very low (2.35). Students from Figure 4.19 b), and 4.19 c) spent much more time in Pocket Paint with drawing and editing their graphics and assets (see bars bordered in green, 3.73 and 3.38). The reasons is that students from Figure 4.19 a) included their pictures mostly from the camera then drawing them (see bars bordered in in blue 3.42). They made photographs of their own experiments and included them in their games. In addition, students in Figure 4.19 b) spent more time in the web-view (see bars bordered in in orange: 3.73) and testing and playing (3) their games than the other two classes.

Behavioral analysis on gender

While the differences between the classes can be mostly explained because of the different class projects and approaches used, a behavioral analysis on gender is highly interesting. This data were analyzed within this section.

First, the analysis over all classes is presented and next, the classes are compared. In Figure 4.20, all three classes are compared. Slight differences are notable between female and male students in their engagement (0.4), effort, confidence, and interests (0.5), and the highest difference is visible in the parameter “Creativity” (0.6). In general, the higher value is always dedicated to the female students’ behavior.

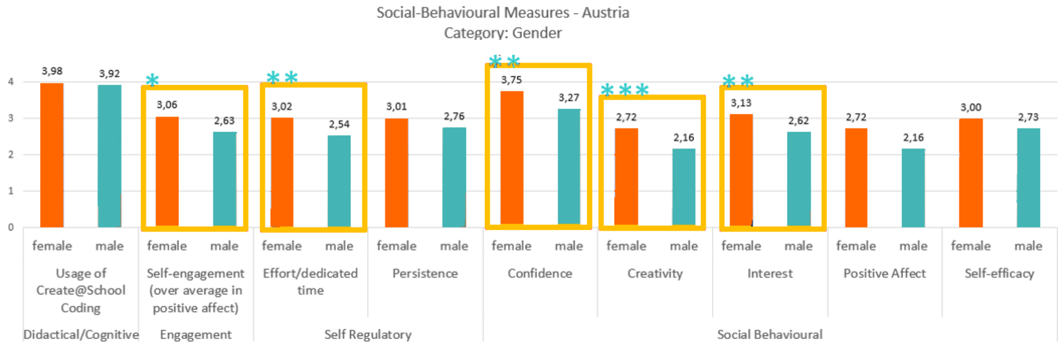


Figure 4.20.: Social behavioral measurements of all classes; filtered by category Gender * Difference 0.4, ** Difference 0.5, *** Difference: 0.6

This means that female students engaged more within the app (e.g., through customization of the app), they spent more time within the app, and most of all they were more creative in comparison to their male colleagues. Figure 4.21 - Figure 4.23 show the gender differences per class. This time each class is described separately.

In Figure 4.21, the "Engagement" parameter shows a significant higher value in those of girls (2.84) than in those of boys (2.06) as well as in parameter "Confidence" (girls: 3.94 and boys: 3.30). This means that the female students in this class used more creation events and spent more time within the app than male students spent. In addition, female students reached almost the maximum value, which means according to the grading score that they had an excellent performance. However, the most interesting parameter is the "Creativity" and "Positive Affect". Here the value for female students is 2.71 and the value for male students is one grade lower at 1.75. This means that the female students used many more of their own camera pictures for their experiments in Create@School than the male students did and they reworked their pictures within Pocket Paint as well (value of "Time spent in Pocket Paint"). Moreover, the program was originally rated much higher by the teacher and they also achieved the learning goal much often (part of the teachers' observation data coming from the PMD). In addition, the value for "Interest" is higher in girls (difference: 0.6) which means that girls spent more time within the app (e.g., with testing the game) than boys. All students in this class showed almost the same value in the parameter "Self-efficiency".

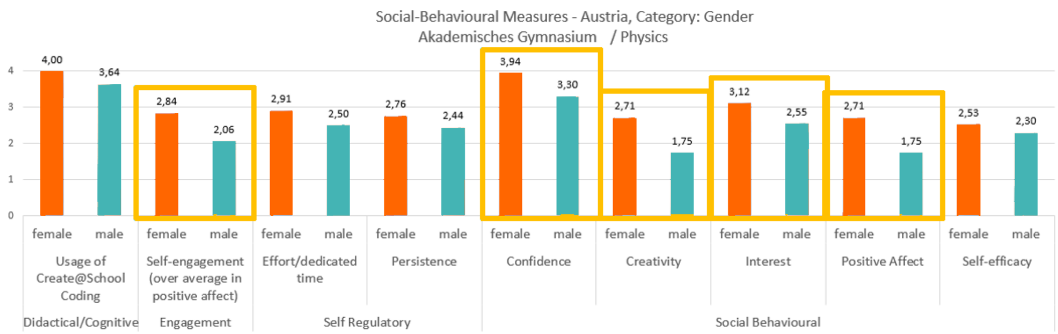


Figure 4.21.: Social-Behavior measures on gender: Akademisches Gymnasium Grade 7/physics.

The class in Figure 4.22 shows less difference between the female and male students. Here, the value of male students in "Usage of Create@School Coding" is minimally higher, and almost reaches the maximum of 4. Here the only significant differences are in the parameters "Effort/dedicated time", "Interest", and "Self-efficacy" (difference: 0.6). This shows that the female students spent more time with either playing/testing their games or for searching for other games in the web-view (e.g., searching for the games of their team members to merge them together). In addition, female students used more creation events, and their own assets (parameter "Interest"). Finally, in this class the female students seemed to spend more time in the app in total, produced more events per time, and used more improved advanced coding bricks (parameter "Coding Skills"). All students show almost the same value in the parameter "Creativity".

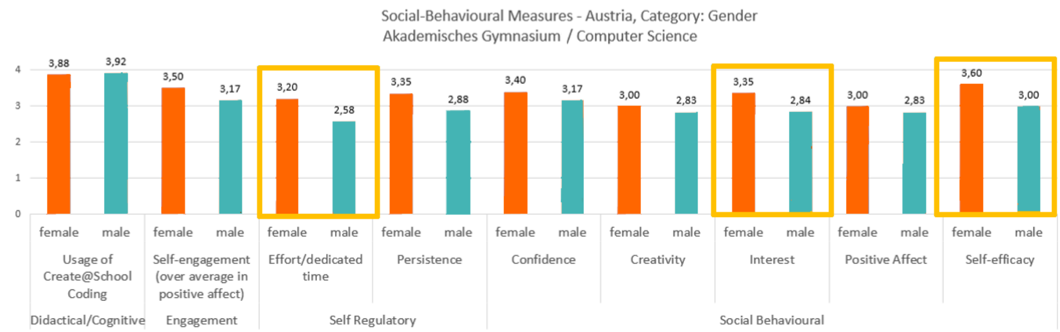


Figure 4.22.: Social behavior measures on gender: Akademisches Gymnasium Grade 12/computer science.

In the third class (see Figure 4.23) few differences are visible between female and male students. Again female students spent more time within the app than male students did (difference: 0.62), and the female students once again showed a higher value in the parameter "Self-efficiency" (difference: 0.66), thus they also used more advanced coding skills in comparison to their male colleagues. In addition, the girls in this class had a slightly higher value in the parameter "Creativity" (thus they added more self-designed assets; difference: 0.5). There are almost no differences in the values of the parameter "Usage of Create@School", "Persistence", and "Confidence" which shows an equal amount of creating and deleting events and reflects their similar coding skills. The reason might be that this class used the pre-coded template, thus the coding bricks needed to fulfil the learning goal in a way that was very similar.

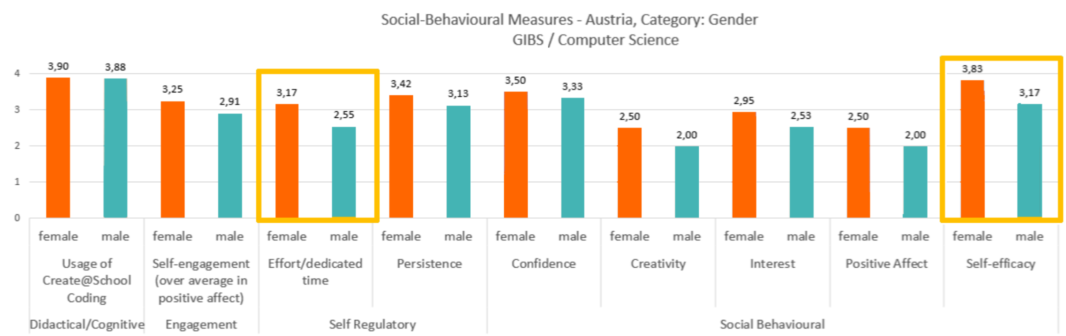


Figure 4.23.: Social-Behavior measures on gender: GIBS Grade 9/computer science.

4.3.2.2. Discussion: behavioral assessment.

For the discussion part, one example program per class and on-site observations are considered as well.

Akademisches Gymnasium, Grade 7 physics

Lesson plan (course 19): Students chose one physical experiment in groups of four and conducted it together within the physics lesson. One member of each group made photographs during important stages of the experiment (the roles changed during the experiments). Every program should contain (see Figure 4.24):

- A start screen which contains the name of the experiment and two buttons: “Explanation”, “Experiment” who lead to the next screens
- One screen with pictures that show an automatic animation of the experiments, a “Return” button that leads back to the start screen
- One screen that holds the explanation of the experiment (picture of a handwritten text), a “Return” button leads back to the start screen
- Optional: “Restart” button for the animation

TU Graz has provided the buttons as well as the background.

Learning goals:

- for physics: plan, perform, and document one physical experiment
- for coding: use the principle of broadcast messages to change between the different screens

According to the behavioral measurements per gender, the female students were more creative, they spent more time within the app with testing or in Pocket Paint, and they used more creation events. On-site observations showed that the girls in this class were much more motivated and collaborative than the boys. Most of them worked in groups of two or alone whereas the boys worked more together with the facilitators. For the girls it was important that their game looked nice as well. Thus, they spent a lot of time in placing the objects correctly, testing the game various times, and cutting their photographs until it suited them. Boys, in contrast, mainly focused on finishing the task in time and being the first to finish. In addition, the optional task (to integrate a restart button) was mostly done by boys. Since they only coded in three units, and did not had any time to program at home, the absolute amount of time spent within the app was more or less the same for all in the class. In this class, students did not have the chance to present their work (no values for work defense).

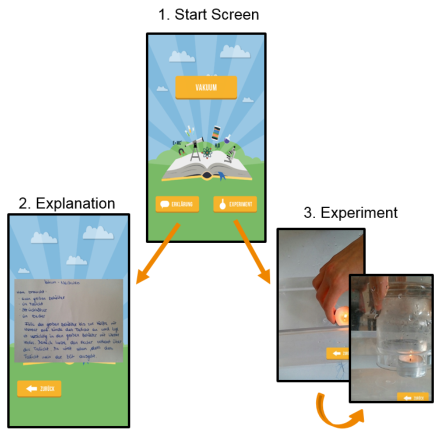


Figure 4.24.: Example game: physical experiment — course 17

Akademisches Gymnasium, Grade 12: computer science

Lesson plan (course 17): Create your own game. The player should either collect something, avoid something, or use storytelling elements (see Figure 4.25 and Figure 4.26).

Every program should contain:

- at least one complete level
- interaction with the player (e.g., inclination sensors or “"When tapped” property, etc.)

Learning goal: use different code bricks from the categories “Control”, “Look”, and “Data” and use object properties.

Lesson plan (course 18): Use the quiz template and add questions related to your school building (e.g., When was it built?). In addition, add the following enhancement to the template: as in a scavenger hunt game, the player should type in a room code (every room in their school building has an unique room code) to unlock the next question.

Figure 4.25.: “Wer ist der Mörder?”: an interactive skill game with storytelling element.

Every program should contain:

- five questions
- the possibility to unlock new questions

Learning goals:

- change the values (variables) within the template to adjust it to your questions
- add an ask brick to enter the room codes

On-site observations showed that the whole class showed great interest in the project and felt very engaged. All the questions were then merged into one big game, which was presented at the open house day in November 2017. Thus, all students of this class were willing to give a great deal of extra effort to finish their games. Female students collaborate much more, e.g., the divided the work between different tablets (drawing, coding) whereas male students worked mostly on one tablet or on their own phones. In addition, boys used very simple graphics or graphics from our media library (see Figure 4.45) whereas, girl’ groups created their graphics on their own. All games created showed a lot advanced functionalities (e.g., using physics bricks, scenes etc.). In the first course, two girls’ groups and two boys’ groups were formed and one group of each gender failed the first learning goal. The girls’ group that did not reach the learning goal, had many problems with the coding and ran out of time (they spent too much time drawing their assets on paper). The boys’ group that did not reach the learning goal integrated too much functionality and deleted something important (without making a copy of the program) and destroyed their whole project. Both problems could be observed as very typical for the gender during the project. However, in the second course (19), more boys than girls reached the learning goal (male: 4/5, female:1/6).

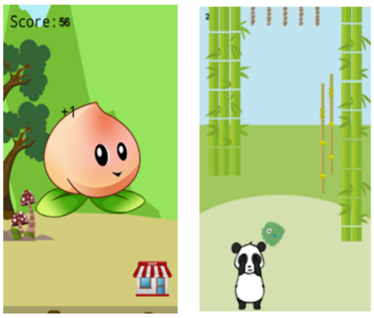


Figure 4.26.: a. “Boom Peach”, a remake of the game “Cookie Clicker” b. “Panda Bouncing”, a remake of the game “Block Breaker”.

The learning goals and a more detailed description of course number 11 are part of Section 4.4.3 (part of the template assessment). In general, again, female students were more precise in organizing their programs, and spent time in searching the right assets. Students were allowed to finish their games at home. Only one girl did so.

To summarize, these results let us conclude that the social behavioral and self-regulatory aspects are the most important for female students. Thus, for them it seems to be most promising to allow them to be creative, i.e., allow to create their own assets, to edit or draw them in Pocket Paint, and to customize the app to their needs. This is consistent with the findings in Section 2.4.2.3 and values in the parameter “Effort/dedicated time”. The task of the first class (record a physical experiment) was not that difficult and very small (adopted for a 3 units course). Nevertheless, it was important for all girls to create “nice” programs whereas boys focused more on the completion of the task (which was observed in many classes during the project). In general, the used more creation events. Girls formed groups more naturally in two classes (either it was individual work, or teamwork) and they split their work more efficiently. Most of the time one girl was working on graphics and the other one was coding (the facilitators encouraged them to change roles) whereas in boys’ groups one boy often took over the program and added fancy functionalities on his own whereas the other group members were often excluded. This is also described in related literature (see Section 2.4.1).

4.3.3. User Experience (UX) with Create@School.

The goal of the qualitative and quantitative research was to collect students’ opinions about the user experience (UX) of the Create@School app and the courses in which the app was used. Data have been collected with open and closed questions to gain a deeper understanding of the experience evaluation. Therefore a qualitative content analysis has been performed by developing categories according to Mayring (Mayring, 2014). The questions were very similar to those asked during the Feasibility Study (see Section 4.2.3). The questions were added at the end of the AttrakDiff survey given to students after their last unit. Overall, a total of 131 students filled out feedback forms in Austria, (63 male students, and 68 female students). The evaluation is part of the NOLB Delivery 5.4 (Spieler and Mashkina, 2017).

4.3.3.1. Usability data: Create@School.

This section presents the usability evaluation of Create@School and the course by answering like/dislike questions.

How was your experience with Create@School? (very good — very bad)

Figure 4.27 illustrates students’ opinions per answers. The percentage of female students (orange) who rated the experience as “good” was 29% (38), and the percentage of male students (cyan) account for 21% (27). The answer option “bad” was chosen by 13% of the girls (17) and 14% of the boys (19). In general, girls rated the app experience more positively (mean female = 2.63) their male colleagues did (mean male = 2.37, but not significantly, t-test: p = .130, α = .05).

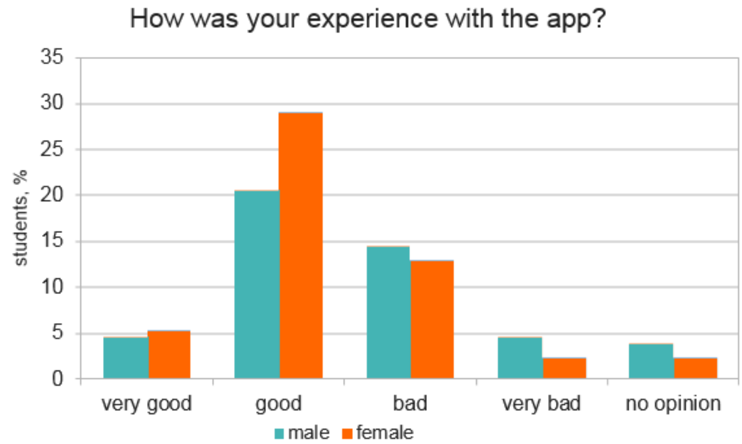


Figure 4.27.: Distribution of the answers about their experience with Create@School.

To clarify the motivation for these answers, it is necessary to take a closer look at the created categories of the open-ended questions:

What did you like the most?

The answers of the participants describe their positive impressions about the Create@School experience. Answers could be classified into five different categories: “working process”, “the app”, “the results (their game)”, “organization”, or “others”. The distribution of the answers among the categories can be seen in Figure 4.28. This time a Mann-Whitney U test was performed.

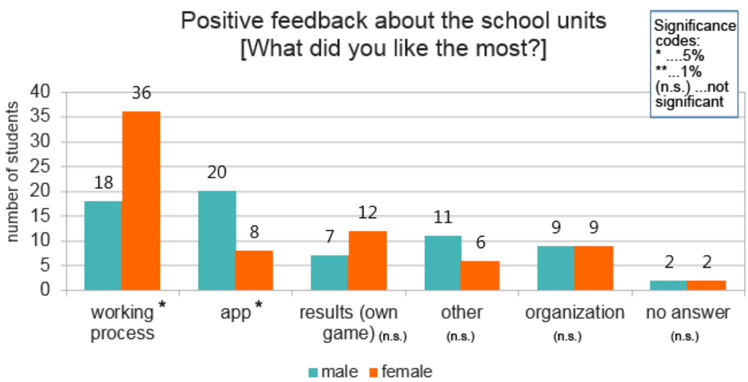


Figure 4.28.: Categorization of the positive impressions during NOLB. Note that one answer could contribute to two categories, for instance the feedback “the finished products + the facilitators” contributed to both, “results” and “organization” categories.

The category “working process” contains feedback about preferred actions (e.g., I like to program, to play the game, to design), or properties of the school units (e.g., having a freedom of choice, that we could be very creative). The largest statistically significant difference between the responses of male and female participants was observed for this category, namely 26% of the girls (36) and 13% of the boys (18) provided positive working process related feedback (z = 2.402; p = .016, α = .05). The category "app" contains the answers mentioning the experiences directly connected with the app itself, e.g. "the simplicity of the app", or "the different effects and backgrounds". Male students seemed to be significantly more positively impressed by the app than girls were (z = -1.969; p = .049, α = .05). A number of 8 girls and 20 boys evaluated the app qualities as positive. Some students were especially satisfied with the results of their work or the concept of game creation, and the category "results (own game)" contains this type of feedback. For instance, "the results and how everything turned out", or "I liked the idea of creating a game" demonstrates this feedback. Of the girls, 9% (12) and 11% of the boys (7) were satisfied with the results or their personal game. The answers from the category "organization" highlights feelings about how the unit was structured. This includes if students were enjoying teamwork or solving the problems on their own, the presence of external people (facilitators of the workshop), usage of tablets during the school units, etc. It was noticeable that students in the younger age group of 12 - 14 years old were very excited about using the tablets during the school units. Typical responses for this category were, e.g., "to work in a team with your friends", "when you (the facilitators) explained to us how the app really worked", "that we had our own tablets". The same number of the male and female participants' feedback falls into the category of "organization" with nine responses from boys and from girls. The replies that could not be clearly classified into any of the described categories above were summarized into the "other" category. For example, answers with no clear message, or containing feedback connected to a particular game or classroom setting, as well as such responses as "nothing" or "everything". There were two such responses from male students and two from female students.

Next, we will take a closer look on the categories "working process" and "app". The category "working process" was divided into six subcategories: "designing", "creativity", "playing", "freedom of choice", "programming", and "other". Figure 4.29 shows the distribution of the responses.

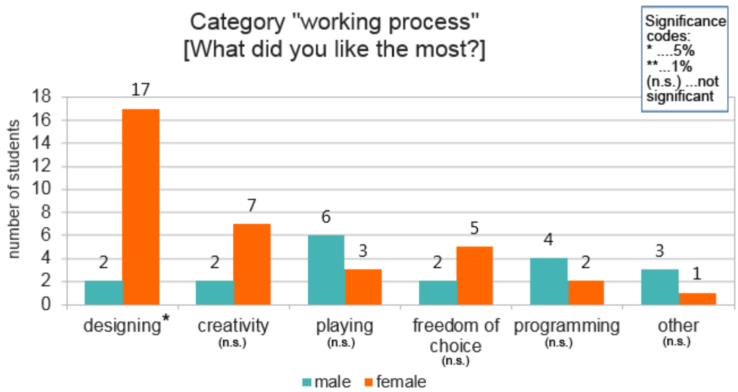


Figure 4.29.: Detailed overview of positive impressions category "working process".

The subcategory with the name "designing" summarizes the feedback related to drawing, taking pictures, personalization of the characters, etc. For instance, "drawing our pictures" or "creating the characters" fall into this subcategory. There were 32% of the girls (17) and only two boys who identified design related activities as the essence of their positive experience (significant: z = 2.151; p = .032, α = .05). The subcategory "creativity" represents the feedback praising the creative side of the school units, for instance, “you could be very creative”, “you could do almost any game you wanted”. A total of seven female students and two male students evaluated creativity as the positive aspect of the Create@School experience. A total of six boys and three girls stated that they considered “playing” as a positive experience. Answers like “you were allowed to play” were typical for this subcategory. No constraints in choice or actions were valued by two boys and five girls. These responses were summarized within the subcategory “freedom of choice”. A representative statement for this subcategory was “That you could do whatever you want”. There were four boys and two girls who enjoyed the programming process itself. Typical responses for the subcategory “programming” were “programming the rocket” or “programming by ourselves”.

The category “app” was divided into four subcategories: “Lego®-style bricks”, “features”, and “design”. The distribution of the feedback can be seen in Figure 4.30.

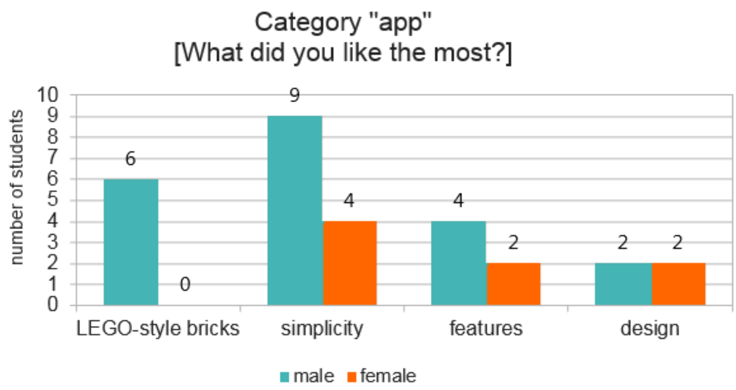


Figure 4.30.: Detailed overview of positive impressions of the category “app”.

None of the girls but six boys stated that they were pleased about the LEGO-style bricks in Create@School. Although the number of answers is too small to be significant, the insights are interesting. The response “brick system” was representative for this subcategory. A total of nine boys and four girls liked the “simplicity” of the app. Feedback included “it was relatively self-explaining!”, or “components are easy to understand”. The subcategory “features” consists of answers like “the different effects and backgrounds” or “variables were available”. Four boys and two girls contributed to this subcategory. Equally, two of each gender liked the design of the app.

What did you like the least?

The answers to the questions were the basis for the negative feedback evaluation. Note that some students responded with everything was fine (6) and some did not give any answer to this question (10). Figure 4.31 categorizes the negative impressions about the app and the school units.

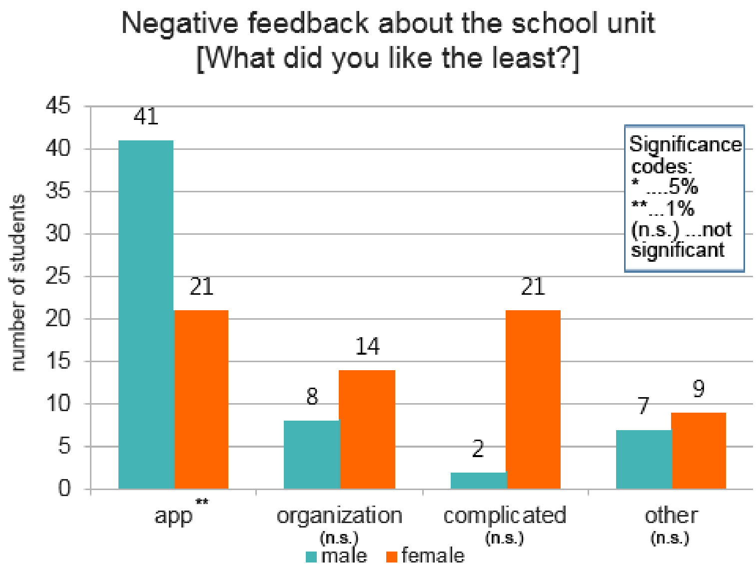


Figure 4.31.: Categorization of the negative impressions about the app and the school unit. One answer can contribute to two categories, for instance the feedback "It was too complicated and it took too long to finish the game" contributed to both the “organization” and “complicated” categories.

The category “app” consists of the feedback related only to the Create@School app, for example, “it looked kind of tacky, complicated, confusing”, “the axis of the screen”, or “it was buggy”. There were 33% of the boys (41) and 17% of the girls (21) who provided this kind of feedback (significant: $z = -3.372$; $p = .0007$, $\alpha = .01$). Responses such as “it took too long to finish up the game”, “introduction of the app was boring”, or “the instructors were not able to explain everything to us” about the unit were summarized into the category “organization”. There were 8 boys and 14 girls who gave this kind of responses. The feedback of the contents of the type “I found some things complicated”, “make it simpler for people that are not technical”, or “we needed a lot of help” were summarized under the category “complicated”. There were 21 girls (17%) and only 2 boys who gave this type of feedback about their experience with Create@School. The replies that could not be clearly classified into any of the categories described above were summarized into “other”, for example, the answers with no clear message, or complaints about the devices and other apps. A breakdown of the category “app” into subcategories can be seen in Figure 4.32.

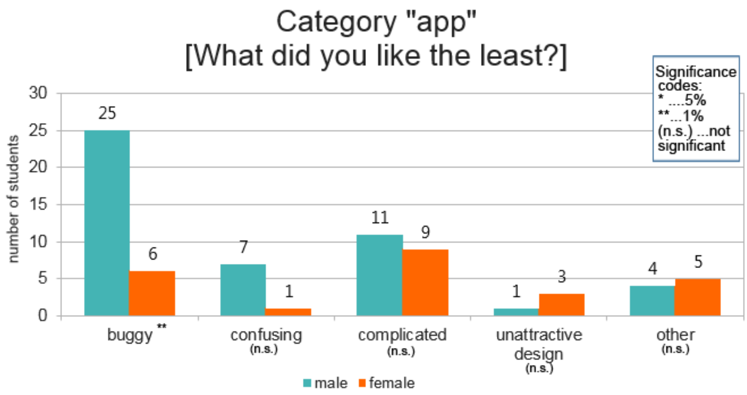


Figure 4.32.: Detailed overview of negative impressions of the category “app”

The subcategory “buggy” contains the feedback about the app behavior that students considered as bugs, responses about slow performance, and mentions of app crashes. 32% of the boys (23) and only 8% of the girls (6) gave this type of feedback about the app (significant: $z = -2.73$; $p = .006$, $\alpha = .01$). The representative answers for this subcategory were “Create@School crashes ALL the time”, “It sometimes stopped, didn’t always work smoothly”, or “it took AGES to load”. Seven boys and one girl stated that the app lacks structural clarity, and is confusing, e.g. “the app totally lacks the structure”, or “some things should be easier to find”. These answers were summarized within the subcategory “confusing”. Nine girls and eleven boys complained about the complexity of the app (subcategory “complicated”). Comments characterizing the subcategory “unattractive design” were, for example “the design is rather boring”. The subcategory “other” contains all other responses that were related to the app, e.g., missing features (“the axes of the screen should be like in Scratch”) or more general feedback such as "it took some time to get used to the program".

4.3.3.2. Further usage: Create@School.

The last question of the feedback form was:

Have you used Create@School in your spare time/outside school? If no, why?

Out of 131 students, 18.27% used the app outside school, four of them being girls and 20 boys. There were 80.85% of the total number of students who did not use the app in their free time. Figure 4.33 shows a pie chart, which is compromised of the reasons why students did not use Create@School or Pocket Code in their spare time.

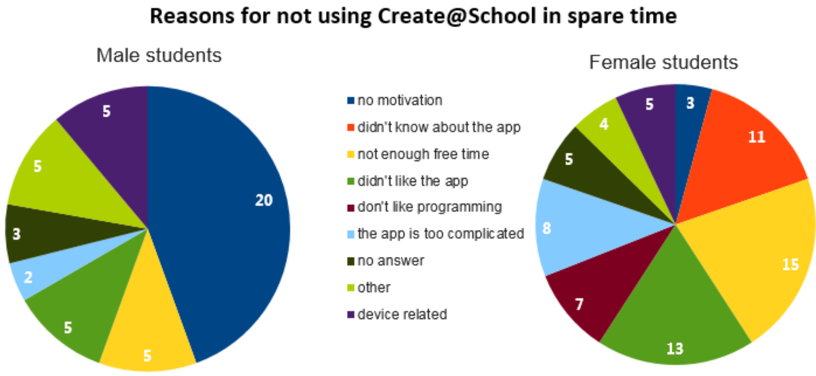


Figure 4.33.: The overview over the reasons for not using Create@School in spare time.

The main cause for 46.6% of the students (44.4% boys and 4.2% girls) was the lack of motivation, typical answers were "I don't want to" or "I have better things to do". None of the boys but 11 of girls answered that they did not know about the app. A percentage of 11% (boys) and 21% (girls) answered that they do not have enough time to use the app, and representative response was "I don't have much spare time". A number of five males as well as 13 female students stated they did not like the app, representative answers for this reason were "I didn't like the app so much" or "I'm not interested in it". None of the boys but seven girls said they do not like programming; typical responses were "Because I am not so technical", and "I personally don't like programming". There were eight girls and two boys who stated that the app is too complicated, for example, "It is too hard and complicated/confusing" or "It would be too complicated to make a game alone". The category named other with 16.63% of all responses contains the answers that cannot be generalized or were not clearly formulated, e.g. "I want to do "actual" coding" or "because it is not my thing". Ten students (five girls and five boys) could not use the app at home, because they either did not have an Android device or were not allowed to install apps on their own.

4.3.3.3. Discussion: usability evaluation.

The results of the UX experiences with Create@School showed many interesting responses and helped the author to shape future workshops. Significant differences between the answers of female and male students can be noticed. For the discussion section, first the results of Create@School are summarized and next they are compared with the results from the UX test of Pocket Code (see Section 4.2.3). Overall, the answers about likes and dislikes can be clearly separated by gender. On the one hand, female students significantly preferred the aspects of the Create@School units ("working process") and mostly related to design activities, but did not mention the programming aspect explicitly. Negative impressions from female students concentrated more on the organization of the units or on the level of complexity, and less on the app itself. On the other hand, male students mentioned app related aspects, like programming and Create@School's simplicity, as positive impressions significantly much more often than the units. Their dislikes significantly concentrated on the app, e.g., that it was buggy. This reflects the complaints of only six girls. Surprisingly, answers from girls also included that they did not like coding at all, whereas this was not an answer given by boys. A very positive aspect could be seen in that 24 students used the app in their free time as well. Here, the answers again are very different between the groups. Whereas girls mentioned most frequently that they had no time or didn't like the app, boys simply felt not motivated to use it. Surprisingly, answers from girls also included that they did not like the coding at all, whereas this was not an answer given by boys.

Results which refer to the units themselves are mostly concurrent with the results of the UX test of Pocket Code in Section 4.2. The comparison between Pocket Code and Create@School shows that this time, the target group mentioned no missing features or functionality. In contrast, with Create@School, the learning aspect of the app was not a topic. However, low performance and high error rate of the app is still a serious issue, but seems to have been more problematic for the male students. Possible reasons are again that boys are more used to utility apps ("tools") and game engines than girls are (see Section 2.4.1.5). With Create@School three times as many students are using the tool in their spare time as well. Reasons for not using the app were once again very similar (no time or interest, device related, etc.).

To conclude, there are statistically significant different aspects which are more important for girls than for boys. In the literature review (see Section 2.4.1.2), the author already described that CS lessons are mostly constructed to suit the interests of males. However, this evaluation shows that the working process and the sequence of the units are particularly important for engaging female students. Thus, for them not only is the tool essential but the learning environment as a whole, as well as the ability to express their own interests, e.g., through designing and creative activities.

Based on these statistically significant results and our analysis of the existing literature (see Section 2.4.2), the author started to design suitable learning environments for girls, particularly focussing on aspects of gender sensibility and awareness. These results are part of the last chapter.

4.3.4. Focus group interviews.

Focus groups are a research method in which the target group discusses a particular topic stimulated by an input and supervised by a moderator (Krueger, 1994; Hennink, 2014). Focus group discussions can thus be used for a wide range of topics, such as product market research. These discussions have the purpose of elaborating explorative content rather than to confirm hypotheses. The evaluation is part of the NOLB Delivery 5.4 (Spieler and Mashkina, 2017).

The conducted focus group discussions for NOLB concentrated on the interests and values of students at risk of exclusion. Moreover, the team retrieved information about the teenage girls' game preferences. Two groups of female students of different ages participated in the focus group after using the Create@School app. The first interview session was conducted with three girls from seventh grade aged 12 to 13 years old in December 2016. These girls worked with the Puzzle template during their arts class (course number 16). They already had previous experience with Pocket Code but did not use the app outside the class. The girls in this course worked individually on their own puzzle game, but formed small groups and helped each other because everyone had the same learning goal.

The second group discussion was conducted with five girls aged 16 to 17 in May 2017. These girls created games with Create@School in context of their computer science class (course no 16). They coded a game from scratch. They did not had any Create@School experience before these units, but they were encouraged to use the app during the Easter holiday period to finish their games.

4.3.4.1. Interview data analysis.

The team only recorded the first discussion. For the second discussion, the moderator took notes only and the answers of this discussion are presented per team (one team with two members: team

1, and a second team with 3 members: team 2). To learn more about our target group, a second focus group discussion was conducted after NOLB by the author herself (see next chapter) with the help of a descriptive content analysis. For the analysis of these data, first, all interview questions are presented, second, the answers of the first group (seventh grade) are provided in Table 4.2, and finally, the responses of the second group (12^{th} grade) in Table 4.3 are given.

- Q1: What was especially difficult in Create@School and why? Improvements?
- Q2: Did something strike you as positive?
- Q3: Would you use Create@School outside school?
- Q4: What is your favorite fictional character?
- Q5: What are your game preferences?

Table 4.2.: Results from the first focus group discussion in December 2016, Grade 7.

# question	Member 1	Member 2	Member 3
Q1	Coding this time was easier for her but she definitely needed the guidance of the NOLB team. She said it would help at the beginning to see how it should look like at the end. Overall, she likes to draw and use her own ideas/graphics but she defines herself as not a professional designer. Thus, it often occurs that her artwork does not look like she expected. She mentioned that in the whole class there are just 2-3 people who she considered as good artists. Thus, it was frustrating to see her kind of "baby" artwork compared to theirs.	It was difficult to understand the structure and the connections between the needed concepts (e.g., to understand what the variables in the templates are needed for). She thought: "What am I doing here?" It would help her to get more tutorials and descriptions of the steps (e.g., what are variables at all, etc.). She suggested that when creating a new game it should be possible to first, choose a topic, e.g., Harry Potter. As a result, the game has all the needed assets from the media library to that special topic, e.g., backgrounds and characters.	She did not mention the scripts at all.
Q2	The template was very easy to expand and especially the upload to the community page was simple. She appreciated the level of freedom of choice (to add own screens, etc.).	It was easy to add new looks and to drag the right bricks. She liked that she was able to add her own start, instruction and end scene images. In the future, it would be great to add sounds and her own voice to the games.	She said that the app was much faster this time. It was easy to use the backpack function of the app.
Q3	No.She considered her knowledge of the app as not deep enough to be able to create something on her own.	It will become boring with time. In "normal games", one has challenges, difficulties, and levels to keep the motivation high.	She said, such apps are for "internet freaks"
Q4	She really likes the movie "Sing" because it states that everybody can do whatever he/she wants to. She likes the story.	Harry Potter ("He is the best"). She likes the whole movie because it displays compassion.	She mentioned Elsa (from "Frozen") with no special reasons.
Q5	Mario Run She preferred "Action" games but she also likes the idea to change the appearance of the surroundings or characters themselves (like in "Sims").	Crossy road She preferred "Skill" games but also appreciated creating something on her own or personalized games (she mentioned as an example to use different kinds of curtains for the puzzle template).	She does not play games but her sister who is one year older plays "Minecraft".

Table 4.3.: Results from the second focus group discussion in May 2017, Grade 12.

# question	Team 1 (2 members)	Team 2 (3 members)
Q1	One girl already had Scratch in the programming course the year before, thus the team did not have any special difficulties using Create@School.	This team had difficulties with the division of the tasks and merging of the scenes (different levels).
Q2	The girls said that it was easy to create their own games from scratch.	This team said that it is relatively easy to use for people without programming experience.
Q3	The girl who did the programming could imagine using Create@School in her free time. She also spent time in the holidays programming the game for the class.	Two of three girls were ready to use the app in their free time; one of them used Create@School but could not succeed due to the diversity of the scenes created by the team members.
Q4	Both girls mentioned movies and characters from mystery and adventure genres, romance as a genre received a negative reaction.	These girls showed a lot of enthusiasm about “Harry Potter” characters and movies. The “Lord of the Rings” movie was discussed in detail, but only one girl called herself a fan.
Q5	This group mentioned diverse games with completely different scenarios,and genres without obvious common patterns, like massively multi-player, online games (MMOG). One girl said that she prefers PlayStation and plays Battlefield. Another girl plays “Candy Crush” in her spare time	Two girls mentioned strategy games in which the player has to build their own environment; one of them mentioned “Clash of Clans” and “Guild of Heroes”, another “Aufstieg von Berk” (at the same time she also mentioned “Smash Hit” as the game she started playing recently).

4.3.4.2. Discussion: focus group interview.

The focus group interview showed, that the template in the first group was received well. On the one hand, they do not describe the units as complicated but on the other hand, they were left with many open questions (e.g., “Why did we change the variable with the name “look number” to value 5?”). Therefore, they would appreciate more general instructions to the template at the beginning or to see how it would look like at the end. The whole group talked a lot about how to make the app more attractive, e.g., by allowing downloading whole media packages specific to one topic, or by allowing more changes/personalization within the templates. All liked the idea of creating their own worlds with self-designed characters and action/skill game genres.

The second group talked a lot about possible themes for game coding lessons. Thus, the movies Harry Potter and Lord of the Rings were discussed as well. In addition, they also mentioned that they prefer to customize the gaming world and adapt it to their needs. Many of these ideas have been considered for the Adventure RPG template, which was developed in April 2017. In fact, Harry Potter still seems to be an interesting topic for the youth (2016 was the year of 20th anniversary of the Harry Potter books). That is why the theme choice for the first Girls’ Coding Day in Summer 2017 was “This is where the magic happens!” (see Section 6.2).

4.3.5. Evaluation of pre-defined templates.

For the assessment, two templates out of 13 (see Section 3.4.2.2) are evaluated in more detail. First, the Physics Simulation template is analyzed in its ability to test the efficiency of the templates in general and to compare games created based on a template with games created without a template. The second template analyzed is the Adventure RPG template, which was developed especially for female students based on the focus group discussions (see Section 4.3.4), and the literature review (see Section 2.4.2.4). To begin with, the overall usage of the templates during NOLB will be presented. The evaluation was part of the NOLB Delivery 5.4 (Spieler and Mashkina, 2017) and part of the authors work in (Spieler et al., 2017).

To accomplish this, the data tracked within Create@School (see Section 3.4.2.4) were analyzed. Data were collected between October 2016 to June 2017. The data were scanned for the event "useTemplate" through all pilots. The tracked data show that the event with the name "useTemplate" was logged 419 times. This means 419 games were created based on a template. Furthermore, the event with the name "createProgram" was logged 623 times in all pilot schools, which means these programs were created without the use of a template. The setup of the school units in which the templates were used was very similar throughout all countries. Therefore, the class teaching strategy was broken into three main sections: the Starter, Main Learning, and Extension (plenary). The Starter includes a debug or basic coding activity to support the learning, but it was the Main Learning section where students started coding, predominantly independently. In Austria, the pre-defined templates were used in five coding courses (course numbers 10, 11, 15, 16, and 18, see Section 3.5.2) in computer science, physics, and arts.

4.3.5.1. Evaluation: Physical Simulation template.

The physics teacher at the Austrian partner school GIBS wished to have a Physical Simulation template in order to consolidate the students' practical knowledge about Newton's 2^{nd} and 3^{rd} law of motion. The template was therefore created together with this physics teacher in several sessions between October to December 2016. Table 4.4 provides details to the Physical Simulation template. It was the idea of the teacher to provide an extra level for students who were faster and to use rockets as objects. Before the actual units with this template started, she provided students the content needed and they created their own paper rockets to conduct real life simulations as well.

Table 4.4.: Physical Simulation template

Topic	Physical experiments with mass acceleration. Applying Newton's 2^{nd} law of motion (F=m*a). Playing around with the properties. Extra task: Simulation of Newton's 3^{rd},law of motion (F1 = F2).
Game play	First level: See what happens if you change the force of the first rocket. The first rocket should reach the finish line. In order to archive this, calculate a=F/m. Second level: Create an own rocket and add variables for mass and acceleration to calculate its force. Third level: See what happens if F1 <F2, F1 >F2 and F1 = F2. Answer the question of which of Newton's laws is applied.
Enhancement by users	Change the force (by tapping on the rocket). Calculate the acceleration. Add another object with mass and acceleration.
Learning goal	Learn about Newton's 2^{nd} and 3^{rd} laws. Add an own object and program its behavior.

Preview:

Figure 4.34.: Physical Simulation template

For the evaluation of this template, the data of two classes in Austria was compared. Both times the app was used in physics in seventh grade (12 to 13 year old, see course number 4, and course number 10) with the same teacher and the same amount of learners (25 students). The lesson with the first class was conducted during the Feasibility Study (December 2015) and the second class had their lessons during the Second Cycle (March 2017).

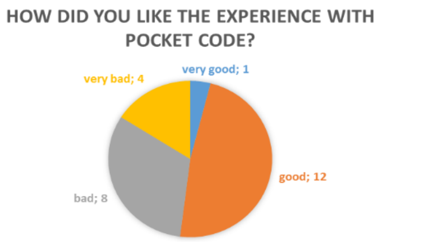
Results: Physical Simulation template

After both units a post-questionnaire was conducted with students and an interview with the teacher was performed, see the results in the Tables 4.5 and 4.6.

Table 4.5.: Physics project with and without the use of a game template

	First class (December 2015)	Second class (March 2017)
Number of units	6 units at 50 minutes: • 4 starter units • 2 programming units	4 units at 50 minutes • 2 starter units • 2 programming units
Theme	Density of objects and liquids Formula: object density <fluid density = item floats object density >fluid density = item sinks	Newton's second law of motion Formula: force = mass * acceleration
Learning goal	• Add an object and let it glide • Set/Show variables to define the properties of the objects • Apply the formula	
Starter unit	Every student started with a new program. One adds one-step to the program in front of the class (see A2). At the end of the instruction units, every pupil had one example level integrated for reference.	The class programs one game together (A1). The finished program was a simple "ping-pong" game, which used physical properties. In the second unit, everyone used their own devices and played the first example level of the template, adjusted the code, and answered the question after applying the formula.
Main learning	Add a new object (personal drawing) and define its movement and behavior by adjusting its density. Apply the formula.	Add own code and a look (picture of self-constructed rocket) to the empty object within the template level and define the object's behavior. Apply the formula. Extra level for students who finished level 2 earlier (to apply Newton's 3^{rd} law of motion).
Learner achieve-ment	**6 of 25 reached the learning goal**	**16 of 24 reached the learning goal (one was absent during programming unit)**
Fazit teacher	The teacher commented that the project was a little bit too difficult for students. Some of them had more problems doing the density program and some were quite fast.	The teacher said that the project was very successful. She found it nice to see the students engaged and working on their programs.

Table 4.6.: Summary of the student's surveys

Questions	First class (December 2015)	Second class (March 2017)
How did you like the experience with the app?		
What did you like the most?	They liked that they could make their own apps and that they learned something about coding.	Most frequently mentioned: It was very straightforward and simple, the programming of the rocket itself was fun, and students liked the possibility to add their own pictures.
What did you like the least? Any suggestions for improvement?	They mostly mentioned that it was confusing at the beginning and that it was very time consuming to have the first simple steps. Improvements: more help functions, tutorials, more tools, less typing.	The most mentioned comment was that the game to create was again seen as too complicated and it was confusing them. Improvements: to add a search function for bricks.
Have you used Pocket Code in your spare time?	None of them. Reasons for not using Pocket Code: no interest, no time, too difficult.	2 used it in their spare time. Reasons for not using Create@School: no interest and no motivation to use it.

Discussion: Physical Simulation template

Based on the results from 2015, the whole course has been adapted to better suit the needs of the students. For instance, students received a more general starter lesson about the application itself. Therefore, the template integrated the idea to let students first change the existing code, thus understanding the overall concept of the game template. In the second step, they had to add a similar object and applying the same concept they learned from the previous level. Therefore, instead of four instruction units, only two were needed. The statistical evaluation with a t-test shows that the experience with the template was significant better than without a template (z -2.539; $p = .014$, $\alpha = .05$). The on-site observations showed that the second class understood how to applying the physical formula better and solved the physics related problem. In contrast, the first class dealt more with programming/app problems and most of them did not reach the point of applying the physical formula. Thus, the students of the second class felt more engaged, were more concentrated, and more of them reached the predefined learning goal as a result. It addition, it was seen as very positive that the template allowed personalization by adding student's own pictures of their self-constructed rockets.

4.3.5.2. Evaluation: Adventure RPG template.

The idea and the content for the Adventure RPG template is based on the results from the focus group discussion with three girls in December 2016 (see Section 4.3.4) and based on the literature review (see Section 2.4.2.4). For instance, all of students who were asked agreed that they liked to play similar games that allow customization; one girl also mentioned she enjoyed playing the "Sims"[84] because she could customize her own characters. The results in Section 4.2.1.3 show girls favorite genre: jump'n'run. However, an adventure RPG genre allows for the creation of more well defined "worlds" and elements of storytelling and narrative characteristics, player character development, complexity, inventory, etc. (see Section 2.4.2.4). According to the literature, these are all elements that are important for the target group (see Section 4.3.4). It has also a mini-skill game integrated, which is the second favorite game genre of our target group (see Section 4.2.1). Table 4.7 provides the details for the Adventure RPG template.

Table 4.7.: Adventure RPG template

Topic	An adventure RPG game in which the user needs to collect inventory to build a project. In addition, the user can create his or her own game character at the beginning.
Game play	Create your own character and choose between different players. Choose a project and collect items you need in order to build it. Collected items are in your inventory. You can drop them again by dragging them to the dustbin. The inventory is restricted. You can only carry 13 items at once. The dustbin will turn red if your backpack is full. Finalize the project by tapping on the checkmark symbol. In the last scene you first mix the different colors and then add them to the painting.
Enhancement by users	Add your own project and items to collect. You should have at least three parts that can be mixed together in order to get a new part (e.g., wood, nail, hammer to make a ladder, which is used to . . .). Mix the parts to build your project.
Learning goal	Arts: Learn about the color circle by collecting colors and mixing them.

[84] Sims: https://www.ea.com/de-de/games/the-sims

Preview:

Figure 4.35.: Adventure RPG template with storytelling and adventure components.

The template was used in May 2017 with a ninth grade class in computer science (see course 11) over the period of ten school units. In this class were 12 students (6 male and 6 female). Students had previous experience with Create@School as they had used the app in the arts class in December 2016 (course 7).

Results: Adventure RPG template

One of the goals was to define the general applicability of this template. Therefore, students were asked to randomly choose a school subject from a list prepared by NOLB team and adapt the template to this subject. Students applied the template to the following school topics:

- Music: catch different note parts to compose a whole melody
- Math: catch numbers to multiply and get “127” as a result
- Economics: catch either electricity, light bulbs, or power lines to connect the villages with electricity
- Chemistry: catch chemical elements form the periodic table to build the formulas, e.g., C, H, and 0
- Sports: build the right item for the sport activity, e.g., a ball for ice hockey, rugby, or basketball

- Geography: collect the flags of English speaking countries and mix them with the corresponding continents, e.g., flags of Ireland, England, or India
- Computer science: catch different components of a computer, e.g., CPU, RAM, etc.
- Fine arts: catch material to build a stone figure, e.g., water, cement, or sand
- Nutrition: catch different meals and serve them at the end as a whole dish
- History: build a coat of arms by catching the different parts
- English: catch different chars to set together words in English
- Biology: she was absent in the last unit and did not upload a game

Figure 4.36 presents all created games from the RPG template.

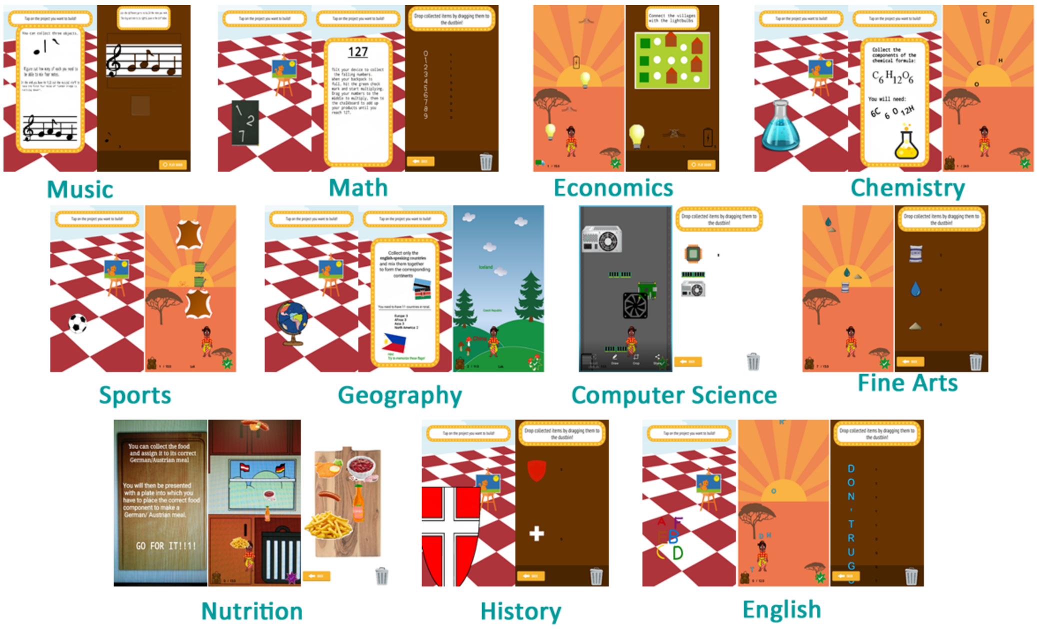


Figure 4.36.: RPG template applied to different subjects.

Before the coding units, the team asked students if they need some graphics from TU Graz design students. Evaluations from the past (see Section 4.2) showed that that students love to use well-designed graphics (see also Literature review Section 2.4.2.3). Three girls gratefully accepted this proposal (for math, arts, and music). See in Figure 4.37 the graphics they needed for their games. They are available within the Catrobat Media Library but only accessible with Create@School accounts.

Figure 4.37.: Graphics for the Adventure RPG template (in Create@School).

After the general instruction unit, students were asked to play the example level of the template to see the general idea of the game. Afterwards, they filled out the first survey to collect their first impressions. Feedback can be seen in Figure 4.38. Because some of the students were missing during the units, the total number of survey participants was 10, six girls and four boys. The sample was too small to determine it statistically as significant.

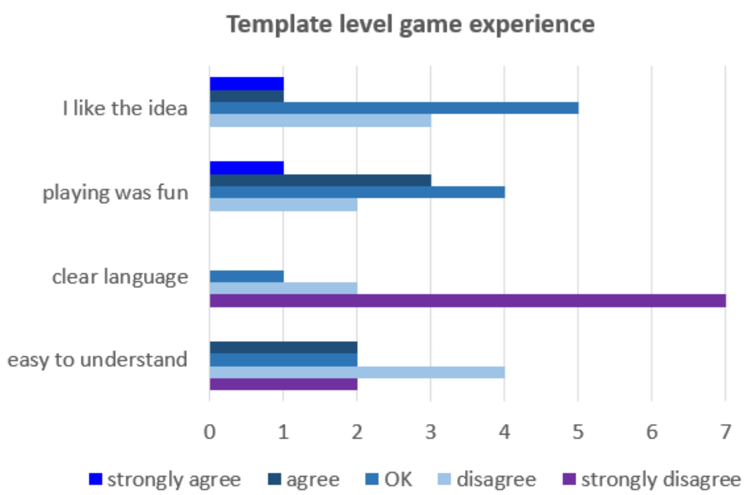


Figure 4.38.: Feedback about the Adventure Storytelling template level trying out the template level.

During the last unit, students were asked to fill out another questionnaire about their experience with the template in general. A number of three students (two girls and one boy) found the template hard to expand. The language of the template was not clear to four of the students (three girls and one boy) (see Figure 4.39). Six students (five girls and one boy) would prefer to work without a template.

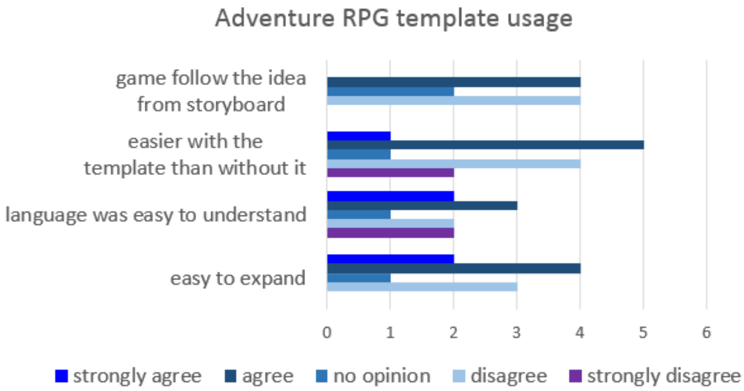


Figure 4.39.: The evaluation of the Adventure RPG template after the last unit.

In addition, at the beginning of the introduction unit, students were asked to fill out a survey about their previous Create@School experience. During the last unit, they were asked the same questions again. As can be seen in Figure 4.40, the results of the second survey are distributed more smoothly and follow a normal distribution pattern.

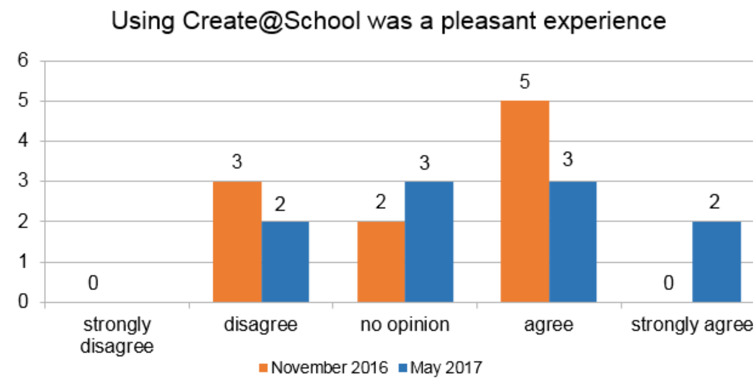


Figure 4.40.: Question after using Pocket Code in November 2016 (without a template) and again in May 2017 (with a template).

Discussion: Adventure RPG template

In the last unit, students presented their games in front of their peers. Only two of them finished the whole template but many completed all the other scenes (instruction, collecting items, and inventory). The template integrates many advanced functionalities (cloning, scenes, variables, drag and drop, etc.) which must be explained first. Before units 3+4 and 5+6, the team prepared a presentation and explained steps students had to consider, theory, and hints. The team was not present in unit 7+8, thus this information was missing. In these units, students tried to finish the last scene, but this one was apparently the most difficult and they were left alone with many questions. With two more units, the NOLB team expects that almost all of the students would have reached the learning goal. As shown in Figure 4.38, after playing the example game of the template, the majority of the students were sceptical if the template could be applied to subjects that were different from fine arts. In the final unit, only one student (a girl) did not change her opinion (see Figure 4.39). However, four students had difficulties finding an appropriate theme for their subject (arts, sports, geography, and chemistry) and they needed help.

On-site observations showed that many female students were more likely to ask the teacher or the team for assistance instead of trying out things on their own. They frequently asked the team for a step-to-step guidance. The team encouraged them to help each other or just to try it out and see what would happen (this align with the literature, see Section 2.4.2.1). Observations showed that they were afraid to "break" the template and two students (one girl, one boy) deleted a needed object accidently. Major problems included a lack of understanding about the structuring of the template and which scripts were essential. One boy was very motivated and created the whole game at home on his own as a new program. He was very disappointed as the team asked him if he could also enhance the template so that feedback could be collected but he was invited to present his own game in the last unit. He strongly customized the design of the template. Most of the male students worked in teams of two and helped each other. Female students worked mostly on their own with all six sitting together. The sample size was very small and in every double unit two students were missing which appears as a problem because they missed the information presented to program the next scene.

To conclude, this template is already a very advanced one. It is perfect for using at school in ninth grade but not without guidance and explanation of the concepts. The template allows customization and can be adapted to different subjects. In order to reach scientific statements about the effectiveness of the template, a larger sample size must be used in the future. Moreover, it cannot be determined whether the template is more suitable for female students.

4.3.5.3. Discussion: pre-defined templates.

In general, all templates were mostly seen as easy to understand, use, and expand. For example, for the quiz template in all countries, students needed time to integrate the first question, but after they understood the overall concept, they added the other questions very quickly. The challenge for developing the templates was, on the one hand, to pre-program the templates in an efficient way which provided students with functionalities to aid the process of building a new game; the templates must explain the important dynamics, mechanics, and aesthetics of a game (like coding how to reward the achievements or how to collect points). On the other hand, students should have the freedom to express themselves in a creative, fun, and dynamic way (e.g., to change images, sounds).

The following challenges could be identified during the use of the templates combined with solutions:

Table 4.8.: Suggested improvement for the templates

Challenge	Solution
students seems to ignore all occurred texts (e.g., in graphics, note bricks)	instead of text, use graphical hints/graphics
the script view in several objects seems to be too complex	put scripts students need to understand/adjust on the top of the script view
naming and amount of variables/ broadcast messages are confusing	refactoring the naming and amount (e.g., more local variables instead of global)
students have no idea how and where to start	mark/rename scenes, object, groups, and looks that should be changed
students find templates too complicated	avoiding long texts in note bricks (just give a small hint) and trying to explain the tasks during the game play
students who are faster have to wait for their colleagues	add extra tasks for them
for every tutorial a description of the shape of a game needed to be integrated and explained how to extend it	explain the shape in one general template, perhaps provide one template that has only the shape of a game (start, instruction, end screen) — could be used to program a game from scratch
provide all templates in portrait and landscape mode	to create game for both modes requires some adoptions in the programs, thus the last four are only in portrait mode.

To conclude, the results showed that the developed game templates encourage learning by doing, allow the expression of one's own ideas, and provide a concept that is easy to understand and to learn. The work to be completed in the future is to estimate the effectiveness of the Adventure RPG template with a greater sample size, create new templates (e.g., for other physical experiments), and eliminate the factors that can give bias to the results.

4.4. Results: Games Created during NOLB

In the last sections, students' experiences with the app and with the workshops were presented along with their tracked behavioral measurements. Other results which were highly interesting are the submitted programs of these students. At the beginning the program statistics are illustrated and later the programs will be evaluated in two ways: Used design elements/pattern (see Section 2.2.1.1) and the functional evaluation (achievement of the learning goal).

First, a game design analysis has been made with the goal of answering the following questions in regard to the literature section: What genre/theme did students use? Are their common backgrounds/characters within the games? Which MDA aspects are fulfilled? From the programming perspective, it is interesting to observe which sensors have been used to control the characters and how many bricks/objects/looks and sound have been used etc.? How were the used objects created (e.g., handwork, from the internet)? Did students integrate the "Shape of a Game" (as recommended by the team, see Section 2.2.1.2) and do all games have a goal? For all of these questions the students' genders are considered as well.

In a second evaluation, the learning objectives defined by the teacher beforehand were measured against the learning outcomes. The results suggest the optimal conditions under which female students best achieved the learning goal.

4.4.1. NOLB program statistics.

This section shows statistics of the NOLB programs on the community website. Tables 4.9, 4.10, and 4.11 summarize all uploaded and finished programs per course (should be more or less one per student or as a group project).

Table 4.9.: Uploaded and finished programs Feasibility Study.

School	Course	# of students	# of uploaded programs
GIBS	1,2	47	19
	3	24	8
	4	25	7
	5	24	4
	6	25	12
Akademisches Gymnasium	12	29	15
	13	21	10
Borg Birkfeld	20	13	6
	21	21	10
		Total	**91**

Table 4.10.: Uploaded and finished programs First Cycle

School	Course	# of students	# of uploaded programs
Akademisches Gymnasium	14	11	-
	15	29	20
		Total	**20**

Table 4.11.: Uploaded and finished programs Second Cycle.

School	**Course**	**# of students**	**# of uploaded programs**
GIBS	7-9	75	26
	11	12	10
	12	25	22
Akademisches Gymnasium	15	29	25
	16	29	23
	17, 18	11	15
Borg Birkfeld	22	19	12
		Total	**133**

All NOLB students had a common username and are represented as a new table in our SQL database. The upload and download statistics from the Austrian students from October 2016 to May 2017 are presented in Tables 4.12 and 4.13.

Table 4.12.: Number of uploads in Austria

Uploads from girls	187 programs	
Uploads from boys	222 programs	
Uploads GIBS	143 programs (53 m, 90 f)	
Uploads Akademisches Gymnasiuim	140 programs (43 m, 97 f)	
Uploads Borg Birkfeld	125 programs (126 m, - f)	
Total	**408 programs**	
Used tags	**No tag**	37 times
	Game	224 times
	Experimental	74 times
	Animation	60 times
	Art	28 times
	Story	22 times

Table 4.13.: Number of Downloads in Austria

Downloads from girls	92 programs
Downloads from boys	252 programs
Downloads GIBS	88 programs (37 m, 51 f)
Downloads Akademisches Gymnasiuim	56 programs (26 m, 39 f)
Downloads Borg Birkfeld	191 programs (189 m, 2 f)
Total	**344 programs**

Although only one class from Borg Birkfeld participated during the Second Cycle, they had the most downloads and many uploads. This class took part in the Galaxy Game Jam event (see evaluation in the next section) and therefore were highly motivated to create a high quality game. Students mostly uploaded more versions of their own games (after every unit) and mostly downloaded their own games or the games from their classmates (e.g., to merge it to one big game). But the download numbers also show some games from the front page of the community website (section most downloaded, etc.) and some remixes (games that have been downloaded, changed, and uploaded again) were present in the upload list. Another important observation is that even through more girls than boys took part in the study, more boys uploaded and downloaded games overall. Boys downloaded almost triple the number of games as girls.

4.4.2. Game design analysis.

Different types of game analysis exist (Arseth, 2003), e.g., players' reports, walkthroughs, discussions, observing others play, interviews of players, game documentation, playtesting reports, or interview game developers. For this analysis, the formal elements are the most interesting (Lankoski and Björk, 2015). For the evaluation of the created Pocket Code games, a form of rapid analysis method (RAM) has been used to identify the different game elements, the game mechanics and the goals they relate to (MDA) (Járvinen, 2007). In addition, special characteristics of the programmed games (theme, genre, goal, etc.) have been analyzed. In detail, the evaluation considered formal game elements, structure, and graphics of the created games (Konzack, 2002; Jesper, 2007). The following analysis has been made separated by gender (see Section 2.2.1.1):

- Definition of the used genre, theme, and goal of the games
- Description of the used MDA: mechanics, dynamics, and aesthetics
- Usage of formal elements, e.g., level of control (interaction), visual design, sound design, restart/end of the game and the "shape of a game" ceremony
- Other design elements: main character, side characters
- Pocket Code specific aspects, e.g., used sensors, amount of bricks, scripts, objects, looks, sounds, and variables

For this evaluation, only the games, which have been created in lessons that used A2 and A3 are analyzed (course numbers 1, 2, 3, 5, 6, 7, 8, 13, 17, 22, see Section 3.5.2). In these lessons, students had a lot of freedom in creating their games and could choose the game elements on their own. In lessons the team followed A1, students were not able to define game genres, themes, etc. on their own (they got a framework that helped them to stick to these good design principles). However, they were allowed to create their own graphics or backgrounds.

4.4.2.1. Results: game design analysis.

Altogether, 77 programs have been analyzed: girls created 37 of the programs, boys 34, and 8 programs were created in mixed gender teams. The number of programs made in mixed teams is too small to perform a significant analysis. Thus, games made in mixed groups are not part of the figures but described in the text. By analysing the games made by boys/girls separately a gender differentiated approaches has been used to trigger a trend. This may can be criticized because "stereotypical" forecasts are often confirmed by it (Grünberg, 2011). In this case it should build gender awareness and help in setting up a more inclusive game design classrooms for female students.

Genre, theme, and the goal of the games

Figure 4.41 shows the used genres/themes of female students and Figure 4.42 summarizes the genre/themes of games created by male students.

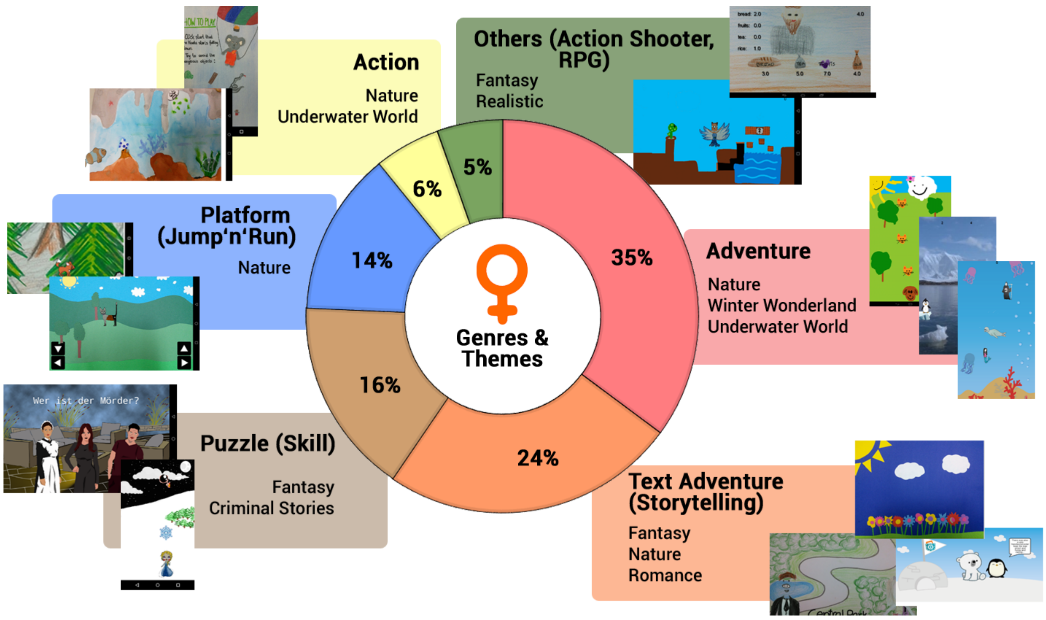


Figure 4.41.: Created games by girls categorized by their used genre/theme.

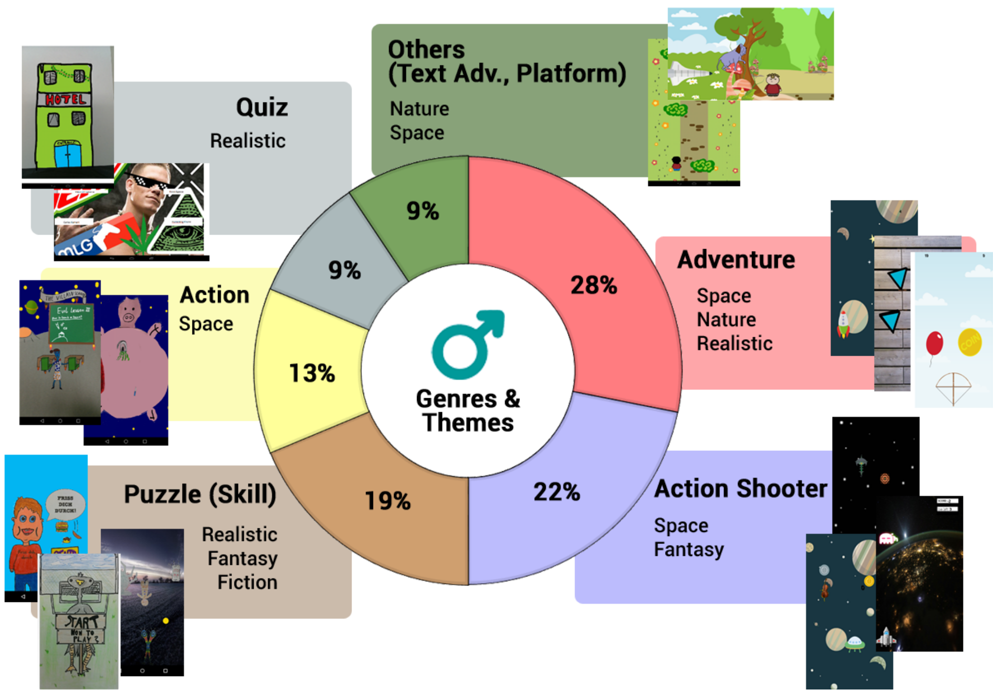


Figure 4.42.: Created games by boys categorized by their used genre/theme.

The games created by girls were mostly adventure games, text adventure/storytelling games, puzzle games, and platform games. None of the games created by girls falls into quiz genre. Games created by boys were also mostly adventure games, followed by action shooter games, puzzle games, quizzes, and action games. Boys, in contrast, created only two games that could be considered in the text adventure or RPG adventure genres. A Chi^2-test has been performed to analyze if there is a significant relationship between the gender and the genre they chose for their games. The results indicated that the relationship between gender and genre is significant $\chi2(71)$ $p = 0.008$ ($\alpha = .05$). A comparison between the adventure genres (text/RPG adventure genre) and the action genre (include the action shooter genre) showed that female students used the adventure genre significantly more often than male students did ($\chi2(71)$ $p = .002$, $\alpha = .05$). The most popular genre among mixed gender teams was the quiz genre (3). None of the created games were based on the strategy genre or the simulation genre (e.g., racing). Female students most often used a nature-based theme (43%) and the most popular theme among male students was space (46%).

A closer look at the underlying goals of the game shows no significant differences between games from girls and boys. Most of the games of both genders had either the goal of Capture/Avoid/Destroy (55%), Collection (11%), Solve (9%), or no goal (e.g. in storytelling games or animations, 18%). “Avoid” is not mentioned in the goals described in 2.2.1.1. Since many games had the aim to avoid, e.g., objects that were falling down, this action has been added to this goal. Territorial, spatial alignment, or building something were never used as goals for any of the games. In addition, 10 of the games used a restart option (12%).

Mechanics, Dynamics, and Aesthetics

Figure 4.43 provides an overview about Mechanics, Dynamics, and Aesthetics (MDA) used by gender. The size of the icon correlates with the frequency of the mentions.

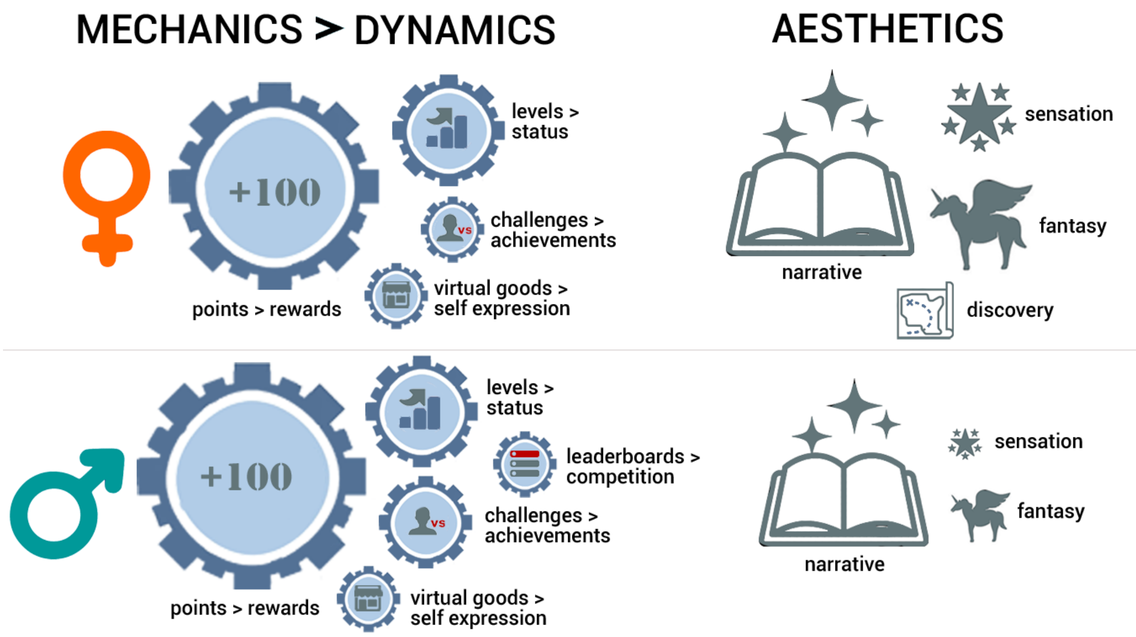


Figure 4.43.: Mechanics, Dynamics, and Aesthetics used by gender.

For mechanics and dynamics, female and male students both most often used points as a reward

(32, female=14, male=17) and levels or lives for status (17, female=7, male=7) in their games (no significant differences). In addition, four of the games created by mixed gender teams have levels integrated. Both groups used virtual goods in the form of a virtual shop to purchase goods in one game, and challenges were used by male students five times and by one female student. Challenges are provided in the form of bosses or achievement of missions through levels. One program created by male students used leaderboards in the form of a high-score list for competition at the end of the game. In mixed groups, levels were used the most often (4) for showing the game status. Together, five games had both points and levels. Gifting has not be used in any of the games.

The definition of the used aesthetic elements is very objective. What one defines for him/herself as sensational could be seen as boring for another user. For the evaluation, the game has been characterized as narrative if, for instance, an intro is provided, the game has a storytelling part, or explanations are provided through the game. A number of 10 games created by girls have this property as well as seven of the games created by boys. In addition, seven of the games created by mixed gender groups are considered as narrative. Sensation is part of the game if the game surprises the player with unusual characters or new concepts (female=7, male=2). The game has the aesthetic of fantasy if the story includes fantasy characters or a fantasy world (female=8, male=4). A game has the characteristics of discovery if missions lead you through a world (female =4). Challenges are part of every game which follows a goal (see previous section). Thus, it has not been added to the graphic (53). Other characteristics like fellowship, expression, and submissions are difficult to integrate in a game made with the Pocket Code app and thus they were not used in students' games.

Formal elements

In total, 40 of 77 games used the "Shape of a Game" (female=23, male=10). Thus, more of the half of the games have a title, instruction, and end screen integrated. Figure 4.44 shows which level of control has been used: animation (female=9, male=3), inclination sensor (female = 15, male = 18), buttons (female = 3), when tapped property (female=14, male =10), or the use of physical properties (female= 1, male = 10). To conclude, both genders used mostly the inclination sensor to control their characters, or the story is continued by the "When tapped" property. Male students used additional physical properties to control their objects, and girls used buttons for movement. The analysis of the Chi^2 test shows that there is a significant relation between the gender and the level of control $\chi 2(71)$ $p = 0.004$ ($\alpha = .05$).

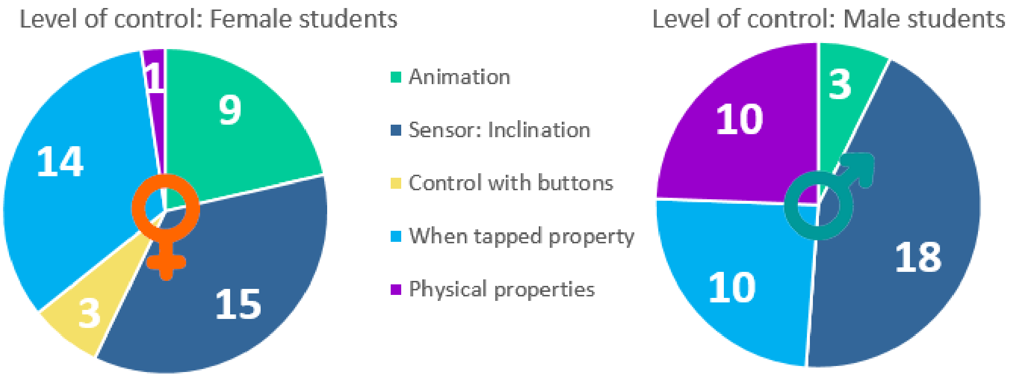


Figure 4.44.: Level of control of games made by girls and boys.

Another interesting question was how the graphics were created. The behavioral assessments (see Section 4.3.2), already show that many of the graphics were either painted in Pocket Paint or imported from the device (see Figure 4.45). The inspection of the programs shows that many of the imported pictures were handmade artwork which were photographed (female = 19, male = 9) or made by students as a whole in Pocket Paint (female = 8, male = 9). Male students mainly used pictures from the internet for their games or utilized the predefined graphics from the media library. In addition, some of the programs made by girls had self-made graphics drawn in Photoshop (3). Games that have photographs integrated showed pictures of the students themselves. In mixed groups, the artwork was either handmade or Pocket Paint was used (4). A performed Chi^2 test showed that overall female students made significantly more graphics on their own (artwork, Photoshop, Pocket Paint) than male students did (Internet, Media Library), $\chi 2(71)$ $p = 0.030$ ($\alpha = .05$).

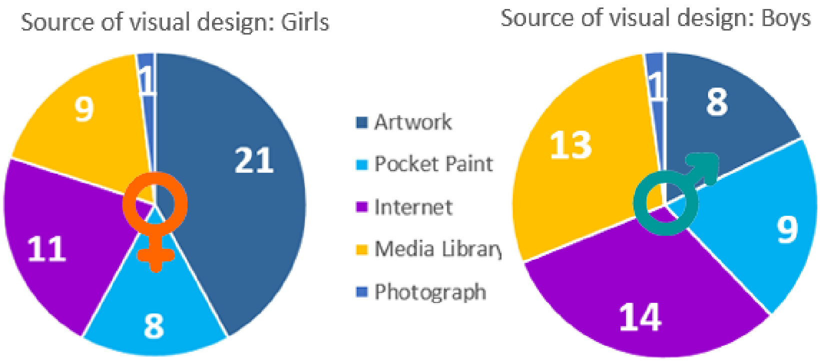


Figure 4.45.: Source of visual design: boys' and girls' games.

In regard to sound design, only 24 games used sounds (31%) in total. 15 games made by boys used sounds (2 used their own recordings, 5 used sounds imported from the device, and 8 imported from the Catrobat Media Library). In games made by females, 10 sound files were found (3 used their own recordings, 1 used files imported from the device, and 6 borrowed from the media library).

Game characters (design elements)

A range of different graphics from different sources were used. For this evaluation, first, the main character and second, the side characters were assessed. Figure 4.46 shows the main characters used in games made by girls. Most games had animals as main characters, like a dog or a penguin (13), or fantasy characters (11), like a fairy, monsters, or a mermaid. None of the games made by female students had a main character which could be characterized as transportation. In contrast, games made by boys mostly (see Figure 4.47) consisted of main characters that represented the category of transportation (12), like spaceships (8), rockets (3), and UFOs (1). A number of six programs made by boys had male main characters. Only one game made by boys had a female character (it is a woman who balances a bucket on her head). Examples for main characters in mixed teams used mainly male characters (4) or animals (e.g., a raccoon or a dog).

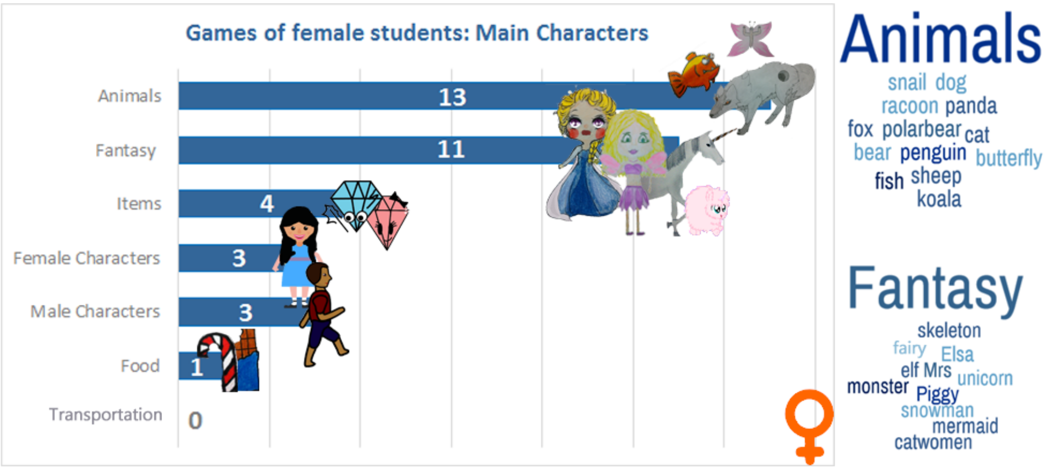


Figure 4.46.: Used main characters in games made by girls.

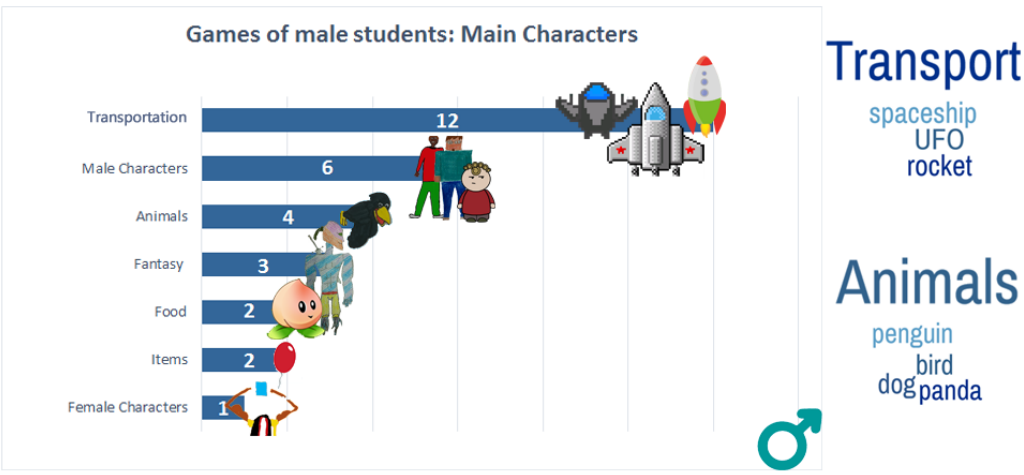


Figure 4.47.: Used main characters in games made by boys.

In addition, a number of side characters are used within the games. A number of 8 games made by females used animals as side characters. Examples are a seal, a cat, or a snake. In addition to fantasy characters (7, e.g., angel, snowman, zombies) many food characters were also used (8, e.g., donuts, carrots, or peanuts). Side characters used by boys were mostly space stuff (13), e.g., meteors, asteroids, stars, or planets, or items (8), e.g., a football or coins. Mixed gender groups used mostly animals (3) for side characters.

A wide range of different backgrounds has been used. The backgrounds used are mostly complementary with the theme of the game, thus they are not described in detail.

Pocket Code specific aspects

Finally, all created programs have been analyzed according to their number of scenes, scripts, bricks, objects, looks, sounds, and global and local used variables (see Table 4.14). This was done by the use of the Catrobat sharing platform's code statistic. The mean value for each element of all programs was calculated and analyzed by using a t-test. The results were also not significant in the means of the numbers of objects in the games ($p = .401$, $\alpha = .05$) but the mean is still interesting: female students used an average of 24 objects in their games, whereas male students used only 11 objects on average.

Table 4.14.: Code statistics NOLB programs.

	female students	**male students**
number of scenes	2.44	2.79
number of scripts	23.97	24.09
number of bricks	120.89	133.63
number of objects	23.89	10.66
number of looks	12.78	14.22
number of sounds	3.16	1.44
number of global variables	3.62	3.81
number of local variables	1.03	1.66

4.4.2.2. Discussion: game design analysis.

The game design analysis shows many interesting issues regarding the different game design patterns of female and male students. The literature review (see Section 2.4.2.4) as well as the play behaviour evaluation from the Feasibility Study (see Section 4.2.1) and the focus group discussions (see Section 4.3.4) builds a coherent structure with the results of this chapter. This means that female participants not only say that they like to play adventure games, jump'n'run games, or puzzles, they also create significantly more games of those kind of genres. In contrast, boys more frequently mentioned sports games or strategy games, but created significantly more shooter games and puzzle games. However, both genders created a lot of adventure games. A possible reason could be that it is the easiest way to program a game in Pocket Code by using the "When tapped" brick or the inclination sensor (see Section 3.3.2), and these kind of games are mostly characterized as adventure games. In addition, girls used several kinds of adventure genres (RPG, text adventure), in contrast to boys, who more frequently used the default adventure genre. The chosen themes of the games show more "stereotypical" choices between boys and girls, but also reflects assumptions from the literature (see

Section 2.4.2.4). Girls mostly preferred nature themed graphics (woods, fields, sea, winter, sky, etc.), whereas boys chose more space, fantasy, and realistic themes which also suited the action shooter genre. As presented in previous sections, female students spent much more time with creating their artwork. This is also visible in the results of Figure 4.45, which shows that girls used significantly more handmade artwork and boys used significantly more graphics from the media library or the internet, or in the design of their main characters (see Figure 4.46 and Figure 4.47). Moreover, the used MDA shows that students of all genders know how to integrate mechanics of points, levels, and challenges for the player, but for female students aesthetics are more important than for male students. Games created by girls used more elements of narrative, sensation, and fantasy than boys' programs.

To conclude, this analysis once again highlights how important it is that female students get enough time to create their artworks and build up the concepts that drive their gaming worlds (see Section 2.4.2.3 and Section 2.4.2.4), whereas the programs of boys are more function-oriented. In detail, boys focused less on creating sophisticated graphics, but struggled more with functional aspects (this is coherent with results of Section 4.3.3) and the code, e.g., they used physical properties (see Figure 4.45) to control their characters which was a relative new feature released in Create@School (feature discovery see Section 2.4.2.5). However, the last table (Table 4.14) shows that there are no significant differences in the scope of the programs (according to used bricks, variables, etc.).

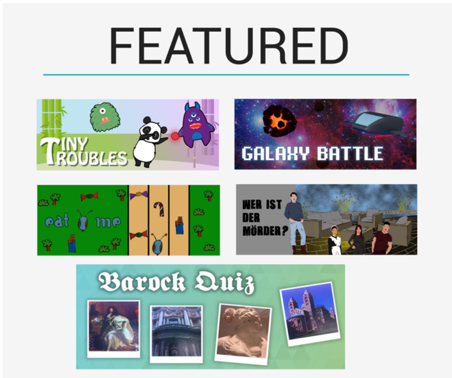


Figure 4.48.: "Tiny Troubles" and "Galaxy Battle" are games made by boys, "Eat Me" and "Wer ist der Mörder" are games made by girls. "Barock Quiz" is a joint program of all games from course number 16 (puzzle template).

Since the number of programs made in mixed teams (8) is too small, no significant analysis of game design in mixed gender groups can be made. The question remains how responsibilities were divided in mixed gender teams. Observations showed that in those teams it frequently occured that only one or two people were responsible for the whole game (this was not dependent on gender).

At the end of the NOLB study, the team chose five games as featured games for the Create@School app. These games are illustrated in Figure 4.48.

4.4.3. Functional game inspection: learning goal achievement.

During the NOLB project, the team rated student's programs in regard to their learning goal achievement. For this assessment, the author analyzed the optimal conditions under which girls achieved the learning goal. Therefore, this analysis should answer the following research question: Which class setting is the most promising for girls to reach the learning goal? The outcome of this analysis will influence Research Question 1 and 2 (see next chapter). The evaluation of the learning goal was part of the authors work in (Spieler, 2018).

For the evaluation, the author considered the subject and grade in which the apps were used, the group composition and size, and differences in the teaching approach (using pre-coded templates or no templates).

4.4.3.1. Methodology for the functional program inspection.

To answer the research questions, a multivariate testing technique for validating the hypothesis with multiple variables was used. With this method, it was possible to determine which combination of variations performs the best. The conjecture was that there are some combinations which are directly related to achieving the learning goal and that they differ between girls and boys. The number of conversions was defined as the number of students who achieved the learning goal, and some independent variables were defined, as described below.

Conversion rate: learning goal achieved

The analysis reported for this analysis investigates the finished program per student. The sample size of finished and submitted programs through all cycles compromises in total 400 programs. (Note: Students who were in the same groups submitted the same projects (including all games, also that created with A1). For the summary, the finished program was counted only once per group.) To ensure the quality of the programs, the team defined learning outcomes per course together with teachers (see Section 3.5.1. Learning outcomes have been defined based on education didactics (Hubwieser, 2007), which expect that a learning goal consists of three parts: action, content, and condition (see Figure 4.49 as an example).

To conclude, each student's work is rated whether the learning outcome has been achieved or not. Thus, the observations fall into one of two categories of a dichotomous dependent variable based on one or more independent variables.

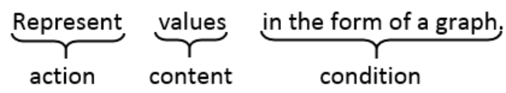


Figure 4.49.: Example: Definition of a learning goal.

Variables

The following variables were noted per student:

Year/grade: For the evaluation, the author summarized "students lower grades" to represent students from sixth grade to seventh grade and assigned "students higher grades" to represent students from ninth grade.

Subject: As already mentioned in Section 3.5.2, Pocket Code was used in physics (amount of students n = 109), computer science (109), arts (184), music (20), and English (49). The learning goal for arts and computer science was strongly aligned with the constructionist learning theory. In these courses, students designed new games from scratch by considering game mechanics, dynamics, and aesthetics (see Section 2.2.1.1). In physics, music, and English, the learning goal was also linked to the subject itself and therefore considered as more interdisciplinary.

Approach: The approaches are the same as described in Section 3.5.1. (A1: template/framework, A2: learning-by-doing, A3: hands-on/freestyle). For this evaluation, A2 and A3 are summarized to A2 because in constrast to A1, both have a more learning-by-doing approach in common.

Group constellation: As described in see Section 3.5.2, the amount of group members varied from 1 to > 4 members per group. A number of 130 students worked individually and 271 worked in groups. Students were allowed to choose their coding partners on their own independently of their gender. Thus, in the study there were 137 females who worked together in groups, 103 males who worked together, and 31 students who worked in mixed groups.

Cycle: During the Feasibility Study and the First Cycle, students used Pocket Code. Within Second Cycle the team used often A1 (pre-coded templates) and all Create@School instead of Pocket Code.

An analysis of the uploaded programs was performed to determine if statistically significant factors, which determined the reaching of the learning goal could be noticed.

Hypothesis

The hypotheses were defined in reference to the literature review (see Section 2.4.2). Thus, the author assumes the following:

- Hypothesis 1: Female students from lower grades (sixth grade, seventh grade) are more likely to reach the learning goal (since they are more interested in ICT/games or are more impartial; see Section 2.4.1.4)
- Hypothesis 2: Female students are more likely to reach the learning goal in interdisciplinary courses (if they use coding in physics, music, and English rather than in computer science or arts; see Section 2.4.2.1)
- Hypothesis 3: There is a correlation between the achievement of the learning goal and female students in connection to the used learning approach (A1 and A2; see Section 2.4.2.1)
- Hypothesis 4: Female students are more likely to reach the learning goal if they work collaboratively in groups with other girls. (see Section 2.4.2.2)

In addition, it was very interesting for us to identify if students benefited from the improved version of the app, Create@School. Therefore, a final hypothesis has been defined:

- Hypothesis 5: Students of both genders are more likely to reach the learning goal in Second Cycle with Create@School.

4.4.3.2. Results: program inspection.

Based on the research design, a chi-square (Chi^2) was performed to test for the existence of a relationship between the variables and thus to evaluate the hypotheses defined. In addition, a comparison to boys is given. The results show also the uplift of the conversion rate (positive or negative in reference to the control group).

H1: Female students from lower grades are more likely to reach the learning goal.

The Chi^2-test was performed to ascertain the effects of the age of female participants on the likelihood that they would reach the learning goal. Students and their achievement of the learning goal per grade is displayed in Figure 4.50. The results found age to be significant predictor for reaching the learning goal in female students, $\chi 2(240)$ p .0009, $\alpha = .01$.). A negative uplift level (-30.61%) indicated that female students in lower grades have a more significant probability of reaching the learning goal than their female colleagues from higher grades. Performing the same tests on male students' programs showed that there is only a marginal correlation between the age of students and the achievement of the learning goal.

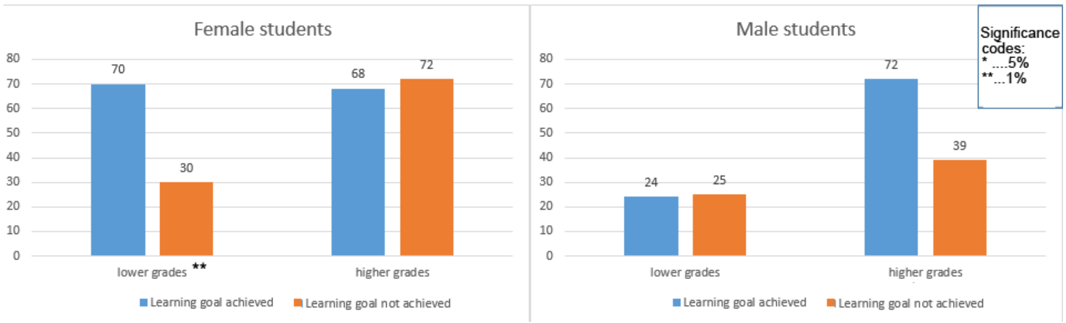


Figure 4.50.: Achievement of learning goal categorized by grade.

H2: Female students are more likely to reach the learning goal in interdisciplinary courses.

Figure 4.51 shows the achievement of the learning goal in the different subjects. Students had computer science only in higher grades. No significant difference had been found in girls between the achievement of the learning goal in computer science or arts compared to the other subjects (not significantly: $\chi2(240)$ p = .075, α = .05). The author performed the evaluation for different age categories, with the same result (not significantly). The same test performed on male students showed that there is a significant relation in reaching the learning goal in arts ($\chi2(160)$ p = .0019, α = .05).

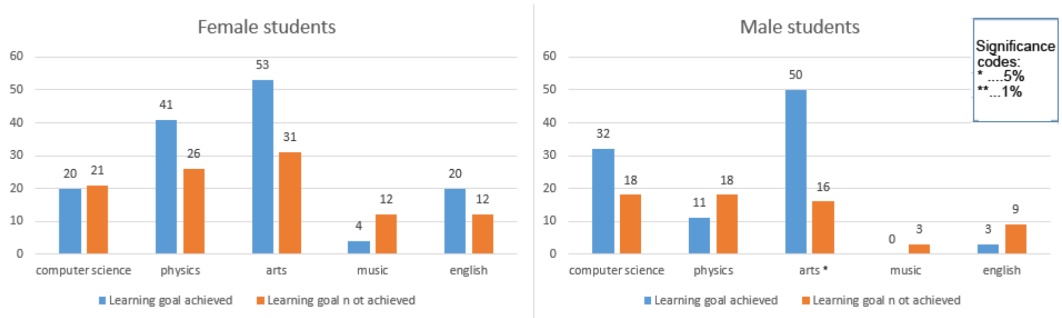


Figure 4.51.: Achievement of learning goal categorized by subject.

H3: There is a correlation between the achievement of the learning goal and the learning approach used

A third Chi^2-test was performed to ascertain the correlation between the used approach and achievement of the learning goal (see Figure 4.52). The results indicated that the hypothesis is true, thus the results are significant; reaching the learning goal was more likely with A1($\chi2(240)$ p = .012, α = .05, Uplift -24.84%). Performing the same tests on male students' programs also showed a correlation between the approach used and the achievement of the learning goal. Here, the results showed a positive uplift, thus the data are significant with a higher probability of accomplishing the learning goal with A2 (free programming) ($\chi2(160)$ p = .023, α = .05, Uplift +35.55%).

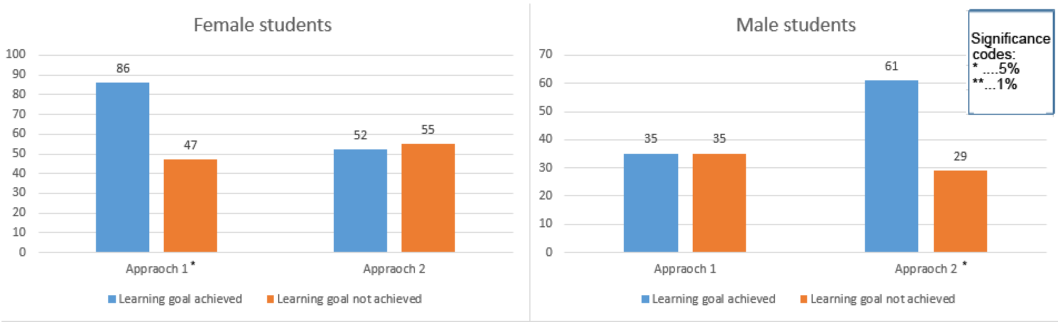


Figure 4.52.: Achievement of learning goal categorized by approach used.

H4: Female students are more likely to reach the learning goal if they work collaboratively in groups of girls.

Next, the goal was to ascertain the effects of the group type on the likelihood of reaching the learning goal. Since only 11 students who worked in mixed groups submitted a program and there are too few conversions for a test, the author only compared students who worked individually with those who worked in groups (groups of girls or boys; see Section 4.53). As a control category, the author used the group type "individual work". The results were statistically significant: $\chi2(240)$ p = .0006, α = .01. Female participants who worked individually were more likely to reach the learning goal than their female colleagues who worked in girls' groups (Uplift -31.52%). In contrast, male participants were significantly more likely to reach the learning goal if they worked in groups of boys ($\chi2(160)$ p = .0006, α = .01, Uplift +69.9%).

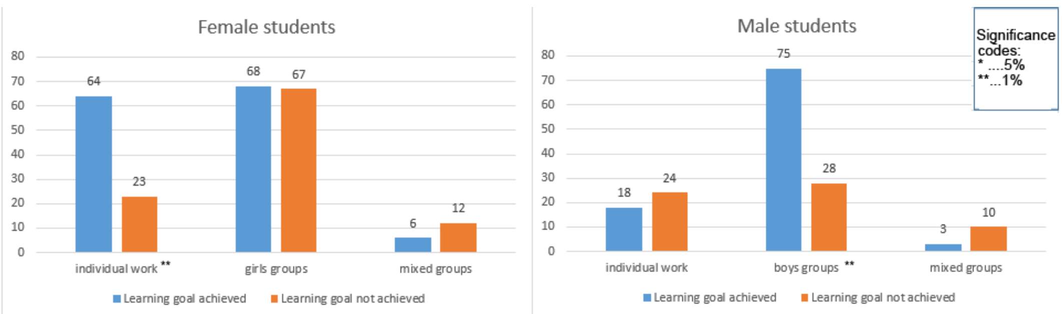


Figure 4.53.: Achievement of learning goal categorized by group type.

H5: Students are more likely to reach the learning goal in Second Cycle.

A last Chi^2-test was performed to see in which cycle female participants would be more likely to reach the learning goal (see Figure 4.54). The results are not significant, showing that girls only marginally likely to reach the learning goal in Second Cycle (but not significantly, $\chi2(240)$ p = .070, α = .05, Uplift +32.52%). Performing the same test on all students and or only at boys shows the same result (not significantly).

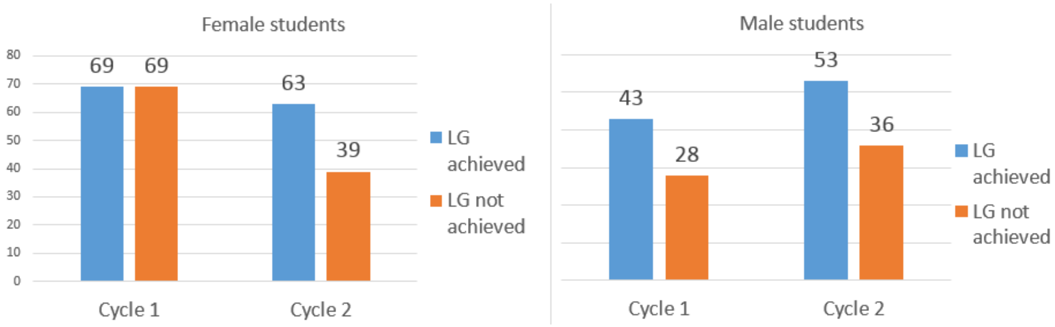


Figure 4.54.: Achievement of learning goal categorized by approach used.

4.4.3.3. Discussion: program inspection.

The aim of this assessment was to validate assumptions from the literature on how to empower female teenagers in coding classes. Based on the significant level and uplift results, the following conclusions can be made: As expected, female students in lower grades were more likely to achieve the learning goal (H1). In addition, in on-site observations, the team noted that they showed much more interest in the app and programming itself; at the same time, they were very careful and insecure in using the app. For instance, they asked questions like “Am I allowed to click on this button?”, or “What am I supposed to do next?”. These findings are complementary with the results in H3 and H5, in which female students achieved the learning goal more easily with pre-coded templates (and marginally with Create@School). However, no significant results showing that the subject influences the program’s completion have been found (H2). All courses were based on game design and problem solving strategies and had the common goal to create or finish a game within different subjects. Because games are known to be a generally promising approach to engaging students (see Section 2.2, it would be more effective to use control groups for testing this hypothesis in the future. This allows us to verify the positive influence of game-based learning with our app in particular. On-site observations in classrooms with female students from higher grades showed that they needed motivation that was more external, and they showed less pride in having their completed program at the end. In contrast to what is known from literature, no evidence was found that female students reached the learning goal more often if they worked in girls’ groups (H4). In fact, the opposite was true. The team observed that female students who worked in groups felt quite stressed during the class. On the one hand, group work was mostly used in combination with A2. Each group member had to create one level, which at the end was merged into one big game. Thus, each of them had different problems or questions, and wanted to finalize her game in time. On the other hand, the team noticed that girls’ groups were not that effective in dividing their work. For instance, it once happened that every group member started with drawing the game assets but no one actually coded. This happened mostly in groups of two (already described in Section 4.2.2. Most teachers noticed this problem and switched from group work to individual work for the second cycle. If students worked individually but all on the same problem, opportunities to communicate and collaborate were provided. Some games, e.g., quizzes or puzzle games (templates), were merged into one big game at the end as well (e.g., merging all the questions in one big quiz).

In a school setting, there are always some limitations to consider. For example, not all students were present in all of the units; either they were absent in the last unit and therefore, they did not submit a program, or they missed too many units. As a result, 70 students were not able to submit a program.

Another limitation of this study was the different amounts of units. In some classes, students had three double units at 50 minutes for finishing their projects; in other courses, they used ten double units. The reason was that teachers from different subjects have different lesson plans they need to cover during class. For instance, an arts teacher can use his or her lessons more easily for coding games than a physics teacher can.

4.5. Results of the Pocket Code Game Jams

This section summarizes the results of the two game jam events performed during NOLB: The Alice Game Jam event (2015) and the Galaxy Game Jam event (2016). The results of the Alice Game Jam event have been already presented (Boulton et al., 2016; Spieler et al., 2016).

4.5.1. The Alice Game Jam event.

The first official Pocket Code Game Jam event were held two times. First, as a test run during the European Code Week[85] from the 12th to the 18th of October 2015 and second, during the International Computer Science Education Week[86] from the 7th to the 13th of December 2015. The game jam aimed to engage female teenagers and introduce them to programming in a playful way (Colley and Comber, 2003). The theme for both jams was as already in the workshops before “Alice in Wonderland”[87]. This topic could be transferred to different subjects like math or literature. In addition, the theme was selected because of the female protagonist, the story’s bizarreness which provides a great basis for creative game ideas.

The first game jam was held using first year computer science students at Graz University of Technology. This event brought insights for the main event in December 2015 and was part of their homework submission. Along with the general theme of “Alice in Wonderland”, our surprise topic was ‘Have I gone mad?’. Moreover, learners could choose a maximum of four of the following diversifiers:

- Using sensors;
- Implementing at least two levels;
- Checking the learning content, e.g., through a quiz;
- Integration of collision detection;
- Using a foreign language.

By the end of the game jam, 200 Alice-themed programs (24% of them from female participants) were uploaded to the community website and can be found by searching for the hashtag #CodeEU . The games comprised a wide range of fascinating and entertaining projects. Based on the feedback of the first game jam in October 2015, the team provided a number of video tutorials. For each tutorial, a sample game was provided. In addition, several Alice themed assets, tutorials, and guideless have been created (see Figure 4.55).

[85] European Code Week: http://codeweek.eu/
[86] International Computer Science Education Week: https://csedweek.org/
[87] Alice Game Jam event: www.alicegamejam.com

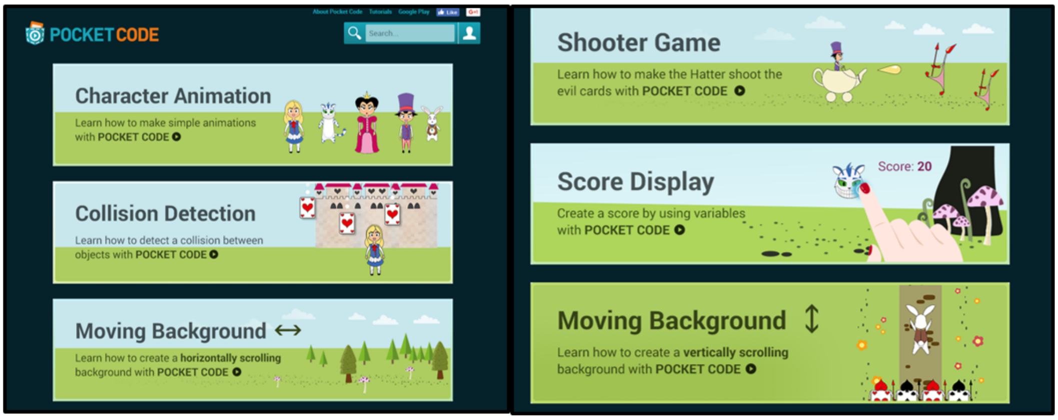


Figure 4.55.: Alice themed tutorials for specific game design steps.

Although the theme and the diversifiers where the same, the surprising topic was new: "Time is running out!". Posters were sent to all of the schools involved in the NOLB project, with further advertising through NOLB partners and social media, such as Facebook and Twitter. In addition, the game jams were conducted together with the MIT Scratch team, the University of Oxford, the British Library, and Game City. During the second game jam event, the following data were collected:

- Raw data from Facebook, Twitter, and the game jam website;
- Number of submissions and details of those submitting, such as country, gender and age;
- A final survey.

Both qualitative and quantitative data were collected: this included demographic data, motivations to take part, the theme, amount of time spent with developing the game, whether they worked alone or in a team, size of team, place where the game was developed (school, home, in transit), diversifiers they had utilized, the time frame, and how they had found out about the game jam.

Data were collected which suggested interest from over 100 countries either through a game submission, taking part in a website activity, or involvement with the game jam through social media. The second time, 95 games were submitted (54.74% Scratch, 45.26% Pocket Code). (All games can be found at the sharing platform with the hashtag #AliceGameJam [88]). Results show that 46.32% participants were female, and 44.21% already had some knowledge in programming languages, like C++, Java, or Python (13), NXT programming (2), in Scratch (3), or Pocket Code (4). The average age of the participants was 17 years. The 95 project submissions were created mostly at home (62.11%) or in schools (32.63%). 75.79% of the participants mentioned in the survey that they liked the theme. The findings indicate that slightly more than half of the submissions (51.57%) were created in small teams (29.47% teams of two; 4.21% teams of three; 17.89% teams that consisted of more than three team members), thus identifying the potential for enabling skills, such as sharing, team problem solving, and cooperation. Games were submitted from a range of different countries (see Table 4.15).

[88] Results Alice Game Jam: https://share.catrob.at/pocketcode/search/%23AliceGameJam

Table 4.15.: Number of submissions for the Alice Game Jam event per country.

Country	# of submissions
Italy	31
India	20
Austria	16
Spain	4
UK	8
US	3
no country mentioned	17

Almost half of the participants spent 2 to 7 days working on their program (44.21%) and 29.47% spent only 2 to 5 hours on programming their games. This shows that the participants were willing to spend extra time outside of school to program their games. Qualitative comments from learners indicated that learners over the age of 16, did not enjoy the theme of Alice due to finding it 'childish', while 100% of those 16 and under really enjoyed the theme felt that it offered many possibilities. Reasons why they participated in this game jam included (multiple answers were possible): "I liked the topic" (23), "I wanted to create a game" (32), "It was part of a school/university activity" (60), and "My friends participated" (7). Surprisingly, nobody chose that he or she wanted to develop his or her ability to code. Only two participants mentioned that they were not satisfied with their outcome. The survey also showed that games were created across different school subjects like math, language learning (German), or chemistry. Therefore, game jams can be adapted to support learning and teaching strategies across different disciplines and obviously do not need be restricted to computer science classes. Learners used a range of diversifiers with sensors, collision detection, and using a foreign language being the most popular. Thirty-seven learners had used only one diversifier; the most popular for those using only one was using a foreign language. From the many submissions of the Alice Game Jam, this section describes three examples. These programs not only followed the game jam theme but also reflect a range of subjects, such as chemistry, languages, and math. Additionally, they consider diversifiers, such as collision detection, the use of sensors, checking the learning content, and integrating the "The Shape of a Game" ceremony framework.

"Sick Alice": ID 5237 (Figure 4.56)

Subject	**Language**	**Gender**	**Age**
Chemistry	English	2x male	17

Figure 4.56.: Best practice example: "Sick Alice".

Description: Alice does not feel well, and it is your turn to help her get healthy again by tapping on the right vitamins. You have to differentiate between water-soluble and fat-soluble vitamins. Be aware: Time is running out.

"Skater Alice": ID 5085 (Figure 4.57)

Subject	**Language**	**Gender**	**Age**
Languages	English and Esperanto	2x female	12, 13

Figure 4.57.: Best practice example: "Skater Alice".

Description: This is a vocabulary game in landscape mode. Memorize the words. You can control Alice with the inclination sensors of the phone. Try to catch the objects from the list.

"Concurso Alicia plantilla": ID 5238 (Figure 4.58)

Subject	**Language**	**Team size**	**Age**
Math	Spanish	>3	16

Figure 4.58.: Best practice example: "Concurso Alicia".

Description: Alice should try to verify math related assertions, e.g., the perimeter of a polygon is equal to the sum of its angles. Every correct answer brings her closer to the White Rabbit and every wrong answer closer to the evil Red Queen.

4.5.2. The Galaxy Game Jam event.

To follow the principles and success of the year before, Catrobat partnered with Samsung to launch the Galaxy Game Jam event in 2016[89]. The duration of this game jam was much longer: from the period of October to December 2016 (again during the European Code Week). The topic was inspired by the partnership with the international company Samsung. This event was an extension of the previous mentioned coding-for-kids' roadshow which took place over nine weeks on the main squares in cities throughout Austria, with morning and afternoon workshops in Pocket Code for school classes. There were several prizes sponsored by Samsung (phones, VR-glasses, tablets). They served as an additional external motivator for the participants and schools that had the chance to win a set of robots for their classes. However, to avoid demotivating any participants, there were no "winners" or "losers". Instead, the prizes were given away at random in a raffle, which even gave beginners the chance to win a device. Special promotion took place at the TU Graz, and all Austrian schools; however, participation was possible for users worldwide. Again, themed assets and tutorials were created (see Figure 4.59). Additionally, in preparation for the Galaxy Game Jam event more beginner steps were explained, e.g., how to install the app and Pocket Paint, how to upload games etc. A lot of promotion work was done through flyers, banners, poster, pre-events, etc. to address a wide range of participants.

[89]Galaxy Game Jam event: http://www.galaxygamejam.com/

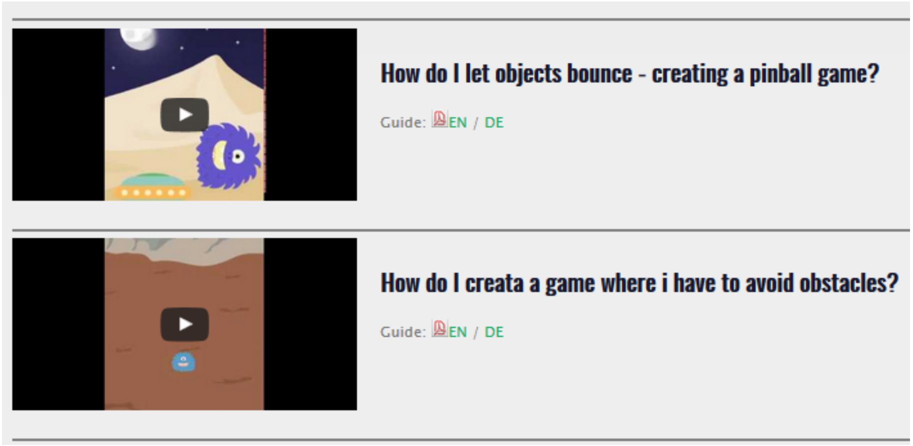


Figure 4.59.: Galaxy Game Jam themed tutorials.

As a result, during the Galaxy Game Jam, 462 projects from 65 countries were submitted, with most submissions from the US (82), Germany (42), Argentina (22), Poland (20), Italy (16), India (17), the UK (13), and Brazil (9). Although promotion mostly took place in Austria, more than 75% of the submissions were made by international participants, resulting in a wide range of different backgrounds and submitted programs. Thus, the game jam was once again a success. 48% of participants stated that they already had programming experience and 77% had done it at home, 14% at school, and 9% at other locations. Games were once again uploaded to the community website[90]. This time, 75% of participants worked on their game alone, and 10% worked in teams (15% in schools). A total of 90% of participants stated that they were very satisfied with their submitted games and 98% mentioned that they liked programming with Pocket Code. The participants were either between 13-16 years old, or older than 16. Reasons the participants gave for attending the jam were (multiple answers were possible) “I wanted to create a game” (55%), “I wanted to develop my ability to code “ (36%), “I liked the topic” (32%), and “My friends participated” (5%).

From the submissions of the Galaxy Game Jam, 10 games have been awarded (5 male, 4 female, and one who preferred not to say their gender) from the Netherlands, Sweden, Germany, and Austria. As an example, one game from a female participant from Austria is presented below.

[90]Galaxy Game Jam page: http://share.catrob.at/pocketgalaxy/

"Fahrt zum Mars": ID 15939 (Figure 4.60)

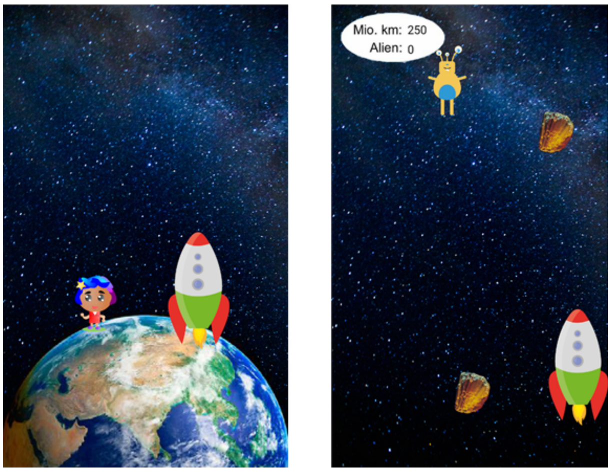


Figure 4.60.: Best practice example (winning game): "Fahrt zum Mars".

Description: The girl enters the rocket and fly into space. The rocket is controlled by the inclination sensor of the phone and has to avoid meteoroids and catch aliens.

However, this time only 8% of the participants were females, with 67% of them having no prior knowledge in coding. This proves that changing the parameters of the event, such as the topic, including more challenges by introducing prizes, or additional promotion has no or a negative effect on reaching this specific target group of girls. In addition, the fact that almost half of the participants already had programming experiences shows that programming novices are not the main target group of Pocket Code Game Jams.

4.5.3. Discussion: game jams as a research method.

Both performed events showed that game jams are an effective way to promote coding among teenagers. The submitted games and data helped us to evaluate the game design behavior of our target group and to gain further insights into the usage of Pocket Code among them. Thus, it shows that this approach particularly enables the long term and regular collection of relevant data for research actions. However, two critical challenges have been identified that question this approach to be of scientific relevance for our specific target group:

1. Low participation of schools/classes:

Although a number of 48 teachers (63% from Austria, 15% Italy, and 17% Poland) showed their initial interest in participating with a class during the Galaxy Game Jam (they respond to our information request about the jam), only students in 12 classes actually submitted a program. This covered mostly schools from Austria (nine classes, two from Italy, and one from Poland) with the highest submission rate in schools TU Graz is already in contact with, like the NOLB partner school Borg Birkfeld (9 submissions) or program submissions which were all not related to the topic. A lot of promotion specifically targeted schools and teachers, but it did not show the desired impact (most games were created at home AGJ: 64%, GGJ: 77%). On the one hand, participants in game jams needed a lot of intrinsic motivation, as a key factor of game jams (Goddard et al., 2014). Thus, in a traditional school setting, there is a need for teachers that motivate students in that topic and provide them with guidance and structure (e.g., introduction, submission day, etc.). Some classes were highly motivated by the prizes (e.g., a class at Borg Birkfeld with only male students) but also very frustrated at the end because no one in the class won anything. On the other hand, most teachers were overstrained by the request to integrate the jam in their school setting. To successfully target schools, it is suggested that helpful material on how to integrate a game jam school lesson needs to be provided, as well as guidance and mentoring during the jam to support teachers who never programmed with Pocket Code before. Help should be provided to give a) support for teachers using gaming in their classrooms for the first time, b) the creation and subscription phase, c) the coding and uploading phase, and d) the evaluation and presentation phase of the games.

2. Low participation of female teenagers.

Whereas the number of female teenagers reached 46% during the Alice Game Jam event, this number rapidly dropped to less than 8% during the Galaxy Game Jam event (number of female participants = 40). A closer look to the answers of female participants shows: most of them came from Austria (30%), 13% from the US/UK or Eastern States (Russia, Ukraine), and 45% come from other countries (Italy, Israel, etc.) or did not mention any country. Most of them were older than 14 years (45%), and most participated alone (53%) or in a class (30%). Only 13% of the participating girls submitted the game as a team. The number one reason for participating were "I liked the topic" (13), "I wanted to create a game" (11), or "I wanted to develop my ability to code" (7). "My friends participated" was only mentioned by five girls and "I wanted to win a prize" was a motivator for three girls. However, 13 girls did not mention any specific reason for participating. On that subject, it would be highly interesting to ask girls (especially in schools) for reasons why they did not participated at all. According to the literature review (see Section 2.4.1), female teenagers need more intrinsic motivators than boys. Thus, providing a challenge and awarding prizes may support a more competitive environment, which is more appreciated by boys and may in fact disengage girls. There is a high risk of inadvertently targeting those who are interested in coding anyway and who are already experienced in coding with such challenges.

In considering what game jams can teach learners about design and programming in creative environments, it is clear that this playful setting provides an inspiring and collaborative atmosphere. Although the data support the evidence that learners can be more motivated through game jams, this concept did not significantly motivate students without any prior knowledge in coding or female teenagers. Furthermore, in the game jam events, the voluntariness of attending (school project/chose freely) must be considered. There is scope for future game jams to further test different topics in engaging all gender. In order to perform successful online game jams on a worldwide scale, a lot of promotion is necessary through partnerships with companies (e.g, with Scratch), as is the adaption of Catrobat materials and processes. After the Galaxy Game Jam event, no more jams were conducted during the period of the NOLB project. After NOLB, the author focused more on game jam events on a smaller scale, e.g., in Summer 2017 a "Girls Day" was performed at the TU Graz with the topic "This is where the Magic happens!", as a result of the focus group discussions (see conclusion chapter). For future considerations, it is still open if the Catrobat team will perform such a game jam event again, but the author will continue to perform initiatives for girls only based on specific topics and providing themed assets or example games (see future work Section 6.2). However, these coding days (or weeks) will not follow the sequence of a game jam, which might be too stressful and not motivating for female teenagers.

4.6. Teachers' Experiences during NOLB

Feedback from teachers has been collected via interviews, on-site observations, and their "diary entries". This means teachers were told to journalize their lessons and send the team their observations, and feelings about the lessons. In addition, we had a debriefing with all teachers of the Feasibility Study in March 2016 (before the start of the First Cycle), and a final discussion round at the end of the project (June 2017) with all remaining teachers who participated in the second cycle. Since the focus of this thesis lies primarily in the female students' experiences, the teachers' view is described very briefly. The number of teachers are assigned in Section 3.3.2.2. The evaluation of the teachers' experience was part of the authors work (Spieler et al., 2017).

The evaluation of the students' surveys and the teachers' feedback were not uniform but mixed throughout both groups (teachers and students), showing points of improvement for the use of Pocket Code at schools and the style of teacher training and support, preparation of tutorials and lesson content, and the backing of the courses. Consequently, only seven teachers (teacher 4, 5, 6, 7, 8, 11, and 13) decided to continue with the project during the cycles (FC, SC). Teachers 1 and 2 (who were both working on the same projects, both with a non-technical background) were disappointed about students' outcomes and expected more advanced games from them. As a result, students felt quite stressed during the course, thus leading to a bad experience for them as well. These two teachers never programmed with Pocket Code on their own, showed no interest in learning about programming, and expected students to discover everything on their own, e.g., search for help online. According to teacher 6, the 15-16 year old students were used to relatively sophisticated games (e.g., World of Warcraft). With Pocket Code, however, they had to downgrade their expectations and hence, their implementation of the games become less exciting. Teachers 3 and 9, both with technical backgrounds, were advanced programmers on their own. Teacher 3 had already programmed with Scratch, but at this time, Pocket Code was not fully Scratch compatible and lacked comfort-functionality, like an automatic collision detection. Both teachers initially had high expectations towards the app but were eventually disappointed as well. However, the remaining group of teachers was highly motivated to work with the app during the further experimental cycles. These teachers had either more than one class at one time (e.g., Teacher 6), several classes in different subjects (e.g., Teacher 7, and Teacher 8), or the same class in several subjects (Teacher 6, 7, and 8). However, they also said that certain parameters need to be established in order to guarantee future success of the units. Teacher 6 evaluated the Pocket Code exercise of his students precisely (Wardell, 2016). His recommendations included to:

1. Design a well-structured pedagogical framework to avoid an overload of technical complexity,
2. Prepare a hardcopy handbook that promotes the overview of all bricks and their functions, along with frequently used brick combinations, and to
3. Limit the project duration to 4-5 double lessons in order to concentrate work towards a deadline and prevent student fatigue with the project.

As a result, the team developed more appropriate teacher guidelines, predefined game templates, and resources for teaching (see previous sections). In addition, more mandatory trainings for all teachers were set, as well as one-to-one trainings to prepare specific units. This should allow teachers to feel better prepared for their lessons by reassuring them that they will be able to conduct their lessons without assistance, regardless of their technical background.

During the second cycle, teachers used predefined templates for their courses (teacher 3, 5, 7 and 8). Teacher 6 again used a package of laminated analogue cards of the Pocket Code bricks, first to translate their composed narratives into coding threads; in a second step, students' programmed a game from scratch. Students in the course taught by Teacher 11 started with their own action games. The feedback and experience of Teacher 6, who did not use a template (but used prepared material for guidance, e.g., tutorial cards), was very similar to his previous courses during the Feasibility Study and students again needed a lot of guidance. On-site observations led to the conclusion that students did not properly define the goals of their games and hence, they lost focus. The rest of the teachers felt more confident during the lessons due the use of predefined templates or the tutorial cards.

Teachers who used the templates with a given learning goal needed to adapt them to their subject. On one hand, they needed more time for preparing their courses, or they needed more individual meetings with the NOLB team. On the other hand, the classroom atmosphere was much more relaxed because teachers guided students more by focusing on the topic rather than explaining complex program structures to them. Most teachers (5, 7, 8 and 13) switched from group or pair work to individual work. If everybody is working on the same problem, everybody could find a different solution for it. In this way, we observed that students were able to support each other by working independently but in small groups, thus feeling more engaged as a result. The observations showed three similar challenges for teachers:

- their confidence in teaching computer science as a subject,
- structure of the course and defining the learning goal, and
- the issue of having enough technical support in the classroom.

In order to give teachers more confidence and guidance, a hardcopy book as proposed by Teacher 6 has been provided (Grandl et al., 2017) and given to all teachers of the SC for free. These could help teachers to feel more confident in using Pocket Code in the classroom. All seven teachers (and Teacher 11) are planning to continue working with Pocket Code after the NOLB project and plan to integrate Pocket Code permanently in their lessons. At the end of the project in June 2017, a final meeting with all teachers from the second cycle was held. The purpose was to finish the project, present results, provide an outlook, and thank the participating teachers and students (who assisted during the lessons). Teachers 4, 5, 6, 7, 8, 11, and 13 appreciated the support promised during the next semesters with different Pocket Code projects. The following is planned:

- Teacher 4 continued to use the predefined templates (Spring 2018)
- Teacher 5 will create a new template for a new physical experiment (Fall 2018)
- Teacher 6 continued allowing students to create their own game ideas but with more focus on fine arts (an example for ideas is to create interactive animations of famous paintings) (Fall 2018)
- Teacher 7 used templates and their games (November 2017)
- Teacher 8 did a game with the Arduino feature (November 2017)
- Teacher 13 did a game with the EV3 Lego feature (but he switched the school and continued without the help of our team)

4.7. Summary NOLB

To summarize, this chapter explained important results of the NOLB project. It discussed all relevant outcomes of the NOLB study regarding female participants and summarized the assessments of the collected qualitative and quantitative data, tracked events, interviews, discussions, and observations. Subsequently, the submitted programs were analyzed in two ways. First, based on the design patterns used and second, regarding the achievement of the learning goal. Finally, the last section of the chapter presented the results of the teachers who participated during NOLB.

The NOLB project has largely achieved its objectives or initiated corresponding measures during the period of the projects cycles: The Feasibility Study helped to identify students' and teachers' specific needs in regard to Pocket Code in order to get information to the target group and provided a list of advantages, barriers, and difficulties in using Pocket Code (see Section 4.2). These results shaped the new app for schools, Create@School, which has been evaluated during the second cycle of the project. The results show that Create@School was positively accepted (see Section 4.3.1) and that students were overall satisfied while using the app. The behavior measurements (see Section 4.3.2) showed that female students were much more creative and active within the app itself in contrast to their male colleagues. Positive and negative impressions through surveys to Create@School (see Section 4.3.3) showed that for male students the app is more important and for female students the workshop setting has a higher priority. The focus group discussion (see Section 4.3.4) helped to receive new insights regarding girls' playing behavior, and finally the evaluation of the templates (see Section 4.3.5) provided feedback for course preparation. Next the author provides insights in students' programs that were uploaded during the NOLB project (design patters and learning goal achievement). Figure 4.61 provides a summary of all NOLB results which are dedicated to female students.

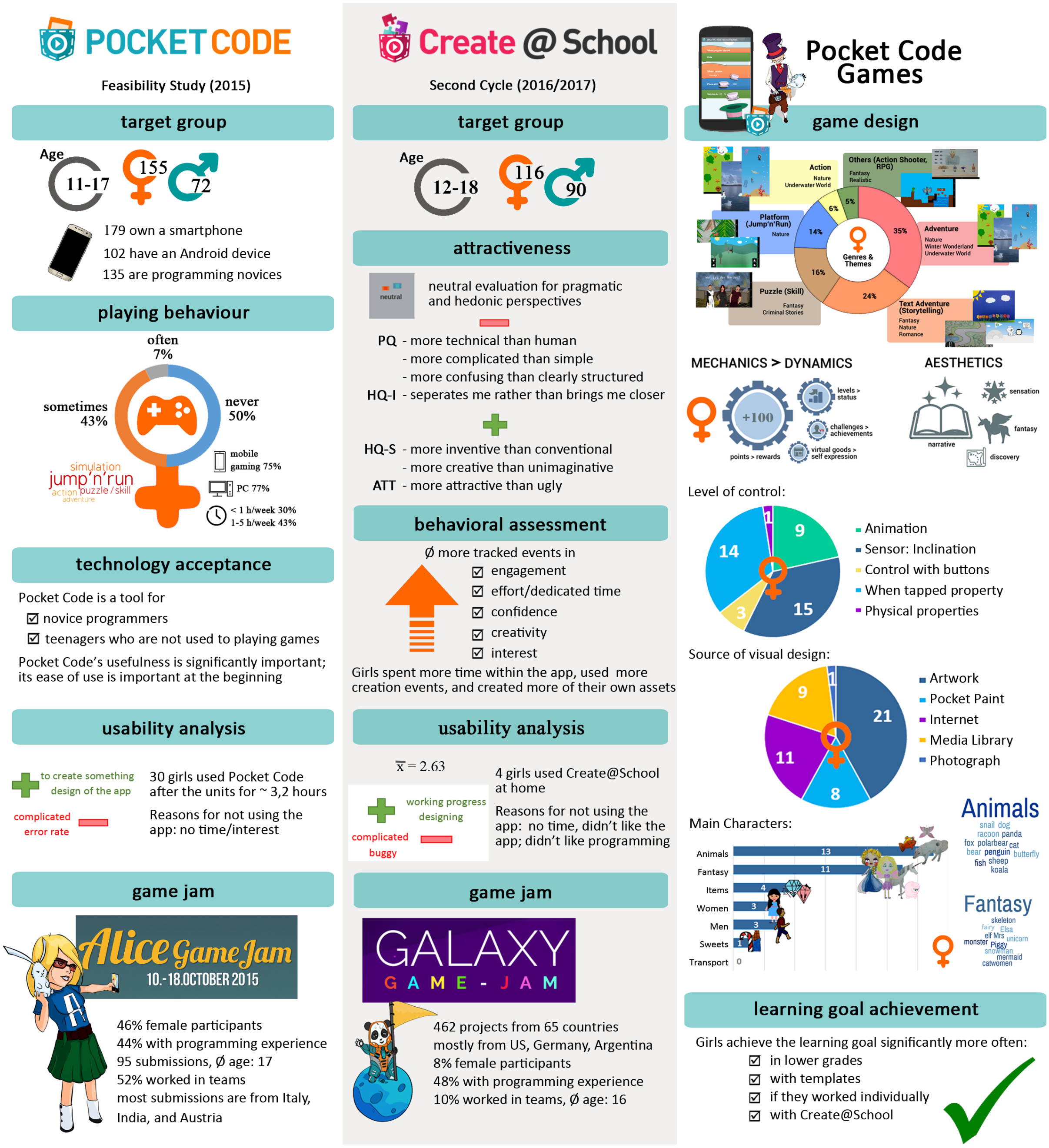


Figure 4.61.: NOLB results regarding female students.

The outcome of NOLB, the Create@School app, was a great opportunity to provide schools with a tailored package of tools and services to help them to integrate coding in their classrooms and to apply the app for interdisciplinary project work. To support teachers in Austria with the upcoming challenge to integrate a basic set of digital education ("Digitale Grundbildung") in secondary schools, three evaluated templates of the Create@School app have been transferred into applicable lesson plans to be used for teachers in regular subjects and they have been provided online for download[91]. In addition, in September 2017 an additional dialog has been integrated in the beta version of Create@School with the option as a teacher or school administrator to request accounts for schools. Since then, 4 teachers from Austria and 36 teachers from outside of Austria requested accounts (mainly from Russia, US, and France). Create@School is still a beta version because it has a required login with restricted accounts. The reason is that this could lead to bad ratings at Google Play from users who are not allowed to log in. Beta version do not offer any ratings. However, the future of Create@School is uncertain. At the end of the project in June 2017, the partner company from Spain, which hosted the BDS client, the Tableau-service, and the SDK for data tracking, went bankrupt. Consequently, the team has had to remove the SDK from the Create@School source code. The analytics tool and the aligned visual dashboards were a key component of Create@School to assess students' performance and engagement during school coding activities. Although the European Commission Committee reinforced the team to create a business and revenue plan for the app (see Delivery 6.7 (Beltrán et al., 2017)) and to search for potential partners who are interesting in adopting this app and its services for school domains, our efforts were not successful.

However, teachers are a very specific target group with special desires for learning tools. On the one hand, they need guides, lesson plans, specific and tailored learning examples, sufficient tools and infrastructures, the technological background knowledge, real time/on-site guidance and mentoring, and reinforcement and motivation. Moreover, most schools in Austria do not have the money to buy and host such tools or services. On the other hand, a school environment has many restrictions like time constraints. Students in a school setting are trained to learn and do things they are not intrinsically motivated by or interesting in. By adopting playful approaches to schools, not all students feel engaged, as one expects them to be. It makes a difference if you place such activities in schools or off school grounds. It matters if students are told to create a game in class for two hours a week over a long period of time and to tell them that they are going to be given an assessment at the end; or if they attend an off-school activity were they feel more attracted to the games and create and design their own apps. It is generally impossible to intrinsically motivate students over a long time period with the same coding activity in schools, but it is possible to spark their interests with a positive experience in coding in just a few hours. To conclude, based on the results of all evaluations during the NOLB project, the author focused her work on how to create playful, engaging, creative coding environments for use in both schools and off-school environments. The goal is to introduce teenagers to the concepts and basics of coding in order to spark their interests and show them new opportunities to improve their lives.

Based on the results of all evaluations during the NOLB project, the next chapter focuses on answering the research questions of this thesis, which summarizes the activities of the author after NOLB and suggests guidelines to effectively support female teenagers in coding.

[91] Learning Lab TU Graz: https://learninglab.tugraz.at/informatischegrundbildung/index.php/pocket-code/

Chapter 5

Results

Given the evaluation of the NOLB project activities (see Chapter 3) and the related literature in Chapter 2, the following Chapter is intended to answer the three research questions, thus presenting the results of this thesis. After NOLB, the author concentrated her research on how to use the NOLB evaluation to not only provide female teenagers with coding activities but to tailor the Catrobat services to their needs. Research Questions 2 and 3 remained open during NOLB and further research was needed. First, the author summarized all results about suitable learning environments for female teenagers (see Section 2.4.2 and NOLB results from Chapter 4 to create a model and guidelines to help to specify coding activities for any workshop contexts. These guidelines should answer Research Question 1: How can we organize playful, engaging, creative, and coding activities to reinforce female teenagers in computer science? Therefore, two courses were organized at our partner school "Akademisches Gymnasium" which follow the PECC model and guidelines. The results have been evaluated by using an experimental design to answer Research Question 2: Do girls-only PECC activities have a positive influence in girls' performance and intrinsic motivation in regard to coding? To answer this, a homogeneous classroom design with female students only was used for the experimental group (the test group). For the second class, the control group was a mixed gender classroom. These evaluations should serve as a case study for girls-only activities. Qualitative and quantitative data have been collected via surveys and interviews. Moreover, the whole units were recorded and observed to see differences in students' behavior, e.g., asked questions, group constellations, etc. The evaluation should not only support the argument for girls-only classroom and evaluate the PECC model but also should provide insights for girls-only activities planned in August 2018 (see Section 6.2).

Subsequently, Research Question 3 has been evaluated: What customizations are necessary in Pocket Code to foster female teenagers in PECC activities? To answer this, a focus group discussion with two classes of different grades was performed. This discussion first, brought insights on our target group and suggested names, designs, etc. for the new app; and second, allowed each student to make proposals for their desired games. Later, these game ideas were analyzed, designed and developed together with university design students at University of Applied Sciences in Graz (FH Joanneum[91]) in Graz. These games served as a basis for the further developed featured games for the new version of Pocket Code with the name Luna&Cat to reinforce female teenagers with tailored PECC tools.

[91] FH Joanneum: https://www.fh-joanneum.at/

5.1. RQ1: PECC — A Model for Playing, Engagement, Creativity, and Coding

Taking cues from the findings from the NOLB evaluations in order to answer RQ1, a new course model — the PECC model — has been developed using four key fields which have been considered as important for such activities, and further intrinsic and extrinsic motivators. The model considers aspects of Playing, Engagement, Creativity, and Coding for an inclusive coding learning environment. These fields are abstracted into an integrated model that provides teachers or facilitators with important elements. In the next chapters, two case studies are examined that demonstrate how this framework can be implemented in a classrooms and to customize tools to validate the models' applicability.

Concepts for coding classes have been applied to teaching in many areas and research for education exists broadly (see (Gibbons, 2013; Shi and Shih, 2015)). Their researches focus on basic techniques and gamified elements, e.g., points or levels, and on game achievements and the use of challenges and competitionsm (Sheth et al., 2012). For this new model, existing research models like PLE/CPL and GDBL have been considered (see Section 2.2.2). The GERD model (see Section 2.4.2.5), as well as important game design elements (see Section 2.2.1.1), and an agile game design process has been chosen (see Section 2.2.1.2). The literature states that girls emphasize the importance of intrinsic and social reward opportunities more than young men who are more interested in placing a higher value on its extrinsic rewards (Microsoft, 2017). Through the literature review in Chapter 2, four important intrinsic and extrinsic factors have been identified. Those were chosen because they influence the motivations towards IT/Coding the most, they serve as important predictors for career choices, or they are important factors to improve coding environments. First, the Tables 5.2, 5.3, and 5.1) presents the key components of the PECC model. In detail they provide a short description of each component, related literature, and how to consider them. Second, each component of the model will be explained in more detail.

The four key components for coding activities are: Playing, Engagement, Creativity, Coding (see Table 5.1.

Table 5.1.: Key components of PECC

Playing	Related literature: (Singer and Schneider, 2012; Li and Watson, 2011) (Chatham et al., 2013; Syamsul and Norshuhada, 2010; Lieberman, 2006) (Chandrasekaran et al., 2012; Kerr, 2006; Wu and Wang, 2012)
	Definition: "Play is freemovement within a more rigid structure" (Schell, 2008) (see Section 2.2.2)
	Provided by: game play, ceremony, developmentand elaboration
Engagement	Related literature: (Papert, 1985; Papert and Harel, 1991; Brophy, 2013) (Owston et al., 2009; Johnson et al., 2016; Dondlinger, 2007) (Corti, 2006; da Rocha Seixas et al., 2016)
	Defintion: a combination of flow, fun, enjoyment, presence, or motivation (Filsecker and Kerres, 2014)
	Provided by: warm up activities, tinkering, unplugged coding, sharing, collaboration, experiments, hands on activities, definitions of responsibilities, tasks, roles
Creativity	Related literature: (Ryan and Deci, 2000; Brophy, 2013) (Kangas, 2010; Kaitila, 2012; Chatham et al., 2013) (Chandrasekaran et al., 2012; IT Manager Daily, 2018; Kahn, 2017)
	Definition: to create something new or valuable
	Provided by: structure design, freedom of choice, asset creation, artwork, tools and practices, stimulate ideas, design, create, available time
Coding	Related literature: (Balanskat and Engelhardt, 2015; Blikstein and Krannich, 2013) (Proctor and Blikstein, 2016; Kahn, 2017; Mannila et al., 2014; Schön et al., 2014) (Brackmann et al., 2017; Robins et al., 2003; Or-Bach and Lavy., 2004) (García-Peñalvo et al., 2016; Lewis et al., 2014; Ildikó Tasnádi and Farkas, 2016)
	Definition: to write simple instructions (algorithm) that can be interpreted by computers to solve problems or create new ideas
	Provided by: visual programming languages, mobile devices, structure coding, system design, new media, guides, starter programs, tutorials, assessment, performance

The four intrinsic motivators are: Interest, self-efficacy, sense of belonging, and fun (see Table 5.2).

Table 5.2.: Intrinsic motivators of PECC

Interest	Related literature: (Sadler et al., 2012; Tsan et al., 2016; RTE, 2016) (Cadinu et al., 2005; Cukier et al., 2002; Gabay-Egozi et al., 2015) (Zagami et al., 2015; Beyer et al., 2003; Unfried et al., 2015) (Ochsner, 2015; Monitise Group Limited, 2014; Master et al., 2016) (Ko and Davis, 2017; Khan and Luxton-Reilly, 2016; Mann and Diprete, 2013)
	Definition: want to acquire knowledgeor learn about something. A feeling of excitement, curiosity or attention to do something (Stangl, 2018).
	Provided by: CS education in schools (6^{th}-9^{th} grade), interesting material and learning goals, templates/frameworks, coding, including real world problems, problem-based learning, game design, interesting games, playing, classroom design
Self-efficacy	Related literature: (Smith and Bowers, 2016; Dasgupta and Stout, 2014) (Cukier et al., 2002; Unfried et al., 2015; Khan and Luxton-Reilly, 2016) (Eccles, 2007; Lockheed and Harris, 1984; Wolber, 2009; Pajares, 2005) (Castillo et al., 2014)
	Definition: refers to a person's believe in their ability to complete tasks and to believe that one's actions are responsible for a successful outcome (Schwarzer and Jerusalem, 1995).
	Provided by: pride, confidence, originality, self- directed learning, computational and problem based thinking, confirmation, performance
Sense of belonging	Related literature: (Stout and Camp, 2014; Baumeister and Leary, 1995) (Walton and Cohen, 2007; Veilleux et al., 2013; Heilman, 1983) (Master et al., 2016; Cheryan, 2012; Good et al., 2012)
	Definition: basic drive for communication, social behaviors, and adaptation (Grossarth-Maticek, 2002)
	Provided by: practical relevance, job clarity, role models/mentors, engagement, self-expression, pride, inclusion, personal goals, values, loyalty, acceptance, fitting, classroom design
Fun	Related literature: (Dee et al., 2009; Games and Kane, 2011)
	Definition: satisfaction and enjoyment, to be happy with the task (not unsure, anxious, and helpless) (Giannakos et al., 2014)
	Provided by: engagement, collaboration, construct, tinker, playing, games, game development, positive attitude,enjoyment, happiness

The four extrinsic motivators are: Inclusion, framework, learning goal, and games (see Table 5.3).

Table 5.3.: Extrinsic motivators of PECC

Inclusion	Related literature: (Braun, 2008; Schmutzhart, 2012) (Vervecken et al., 2013; Vervecken and Hannover, 2015; Szekeres, 2005) (Dasgupta and Stout, 2012; Vervecken and Hannover, 2012; Vervecken et al., 2013) (Vervecken and Hannover, 2015; Sczesny et al., 2016; Horvath et al., 2016) (Chandrasekaran et al., 2012; Craig et al., 2013; Ramos and Rojas-Rajs, 2016) (Formanowicz et al., 2015; Hentsche and Horvath, 2015; Horvath, 2015) (United Nations, 2016; Gleichbehandlungsanwaltschaft Österreich , 2011)
	Description: "improving the terms of participation in society, particularly for people who are disadvantaged, through enhancing opportunities, access to resources, voice and respect forrights" (United Nations, 2016)
	Provided by: warm ups, gender sensitive and aware education, language, communication, motivation, engagement
Framework	Related literature: (Romeike, 2010) (Carter and Jenkins, 1999; Games and Kane, 2011)
	Definition: material that saves as a support or guide (see Section 3.4.2.2 and 4.3.5).
	Provided by: freedom of choice, personalization, customization, creativity, storyboard, MDA
Learning goal	Related literature: (Zichermann and Cunningham, 2011; Romeike, 2010) (Clapper, 2009)
	Definition: see education didactics Hubwieser (2007), a learning goal consists of: action, content, and condition (see Section 4.4.3)
	Provided by: interdisciplinary project work, templates, problem solving strategies
Game	Related literature: (Culin, 1975; Deterding et al., 2011; Shi and Shih, 2015) (Wearn and McDonald, 2016; Pereira and Rodrigues, 2013; Brophy, 2013) (Sampath, 2004; da Rocha Seixas et al., 2016; Nakamura and Csikszentmihalyi, 2009)
	Definition: a voluntary activity binding on certain rules of time and space (see Section 2.2.1)
	Provided by: playing games, game design, game worlds, MDA, genre, theme

The simplified PECC model, defined in Figure 5.1, applies concepts of GDBL, GDFs, GPL and the constructionists approach. It takes four key components for designing appropriate coding activities and two methods (learning by creating a game from scratch, or learning by extending templates) and abstracts them into a model for creating class projects or other kinds of workshops. The defined components are of PECC (1) Playing, 2) Engagement, 3) Creativity, and 4) the Coding) are defined as the most important aspects to consider for coding classes, and consist of tasks to guide teachers and facilitators in their planning, setting up phase, and through the whole course/workshop. In addition, a template for storyboards and an assessment sheet for assessing the produced artefacts was developed (see Appendix A.5 and A.6). These tasks were defined in regards to the NOLB results (see Chapter 4), aspects from gender-sensitive theory (see Section 2.4.1.2), and from the game design theory (see Section 2.2). In addition, the PECC model builds on the NOLB GMTF. The NOLB GMTF provided a frame for the NOLB project to understand and apply mechanics and dynamics as well as to integrate Create@School and the PMD into the academic curricula. The PECC model has been expanded to be more applicable (not only adaptable for schools/teachers, or Pocket Code/Create@School lessons) and further provide applicable proposals to foster inclusion explicitly. The focus now is more on the practicability of coding/game design lessons, thus providing concrete guidelines for teachers and facilitators. To phrase it differently, PECC can be applied to any beginners coding class in all contexts. Thus, for the PECC model, tasks like the definition of the learning goal, game-based methods, and important inclusion aspect have been adopted from the GMTF. In addition, the 4 C's of 21st century learning and CT were also considered for the model (see Section 2.1.4) as well as the idea of unplugged coding activities of tinker and maker spaces (see Section 2.1.4.1). The goal of this model is to foster students' sense of belonging to coding relevant fields, awaken or generate interest for this area, to improve their self-efficiency towards coding, and finally, to bring fun elements to the classroom or a workshop. These are defined as the intrinsic motivators which are especially important in girls: a study (Galdi et al., 2014) added that by directly comparing female students sense of belonging, expectations of success, and stereotype threat, girls' lower interest in computer science can be predicted compared with boys when stereotypes are salient. Extrinsic motivators are considered by the models inclusion of all students. This framework should not reduce students' creativity, but provide defined (learning) goals to strive, and should give rules to consider in traditional games. The PECC model targets programming beginners, students of all genders, teachers of all subjects, and facilitators of coding workshops. It is applicable to different tools/platforms, not limited to Pocket Code.

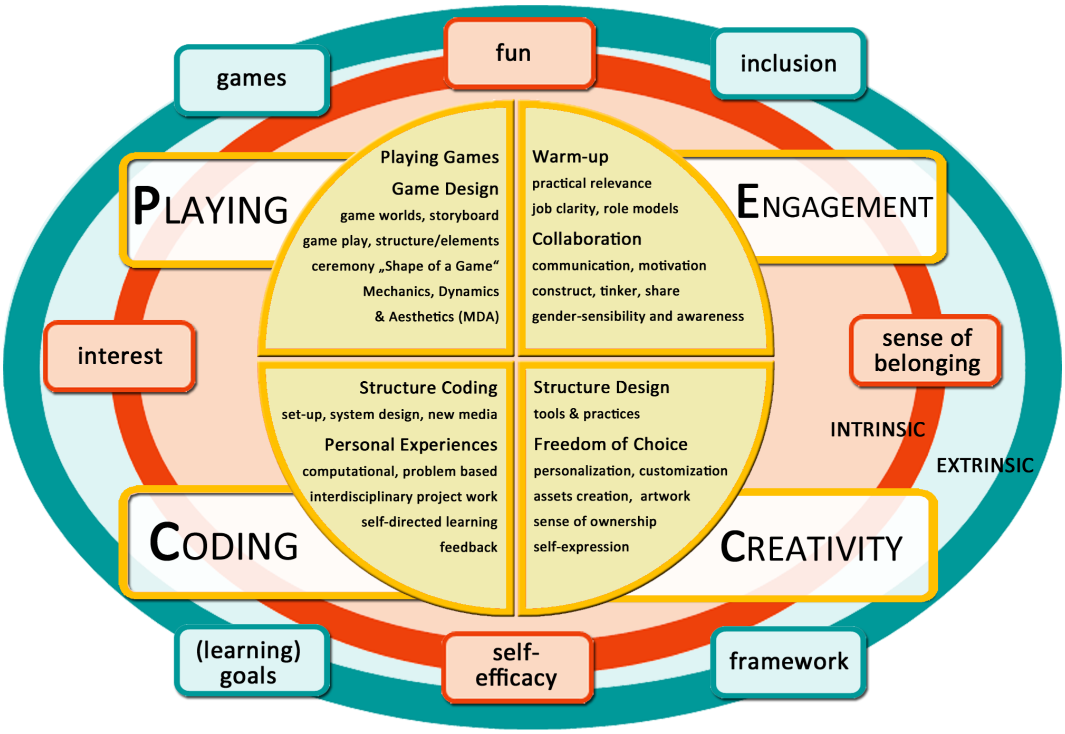


Figure 5.1.: PECC: a model for Playing, Engagement, Creativity and Coding activities.

At the basis of this model, a number of guidelines have been developed (see Appendix A.4) to help teachers/facilitators with concrete steps. For tasks that are not important for workshops outside the school settings, can be left out, or which are especially gender relevant, symbols have been used. The guidelines provide a checklist with tasks that are strongly recommended and some that could be left out (indicated by symbols). Sometimes several tasks to choose from are provided (a, b, c,) and sometimes they are sequential (1, 2, 3,). Moreover, different approaches are considered, e.g., usage of templates/frameworks or starting with an empty program from scratch. In the following section, the components of the model are described and linked to the literature and findings of this thesis. The description refers to teachers and students but that can be replaced by facilitator and teenagers as well.

5.1.1. Playing.

It is well known that students can be motivated by using a fun and playful approach (e.g., (Paderewski et al., 2015; Hulsey et al., 2014; McLean and Harlow, 2017; Denner et al., 2005; El-Nasr et al., 2007; Subsol, 2005)). However, play and games are linked to certain goals and rules (see Section 2.2.1). Hence, teachers must provide these details to help teenagers to stick to good design principles before they actually start to develop their games. Above all, most female teenagers are not used to playing games (see Section 2.4.1.5). Thus, most of them are not familiar with important components of a game, which many boys experience in their everyday life. It is important that students see this process from the eyes of a designer rather than only the player's perspective (see Section 2.2.1.2). The different parts within the field "Playing" are:

5.1.1.1. Playing games.

First, it is important to understand students' playing behavior, what games they like, and what games they play (Nah et al., 2014). Therefore, some questions are useful, e.g., what kind of games students are used to playing, or why they play games (or why not). Second, students can benefit a lot if they are allowed to play similar games made by other users. This is not only important because students may not used to playing but also to show all what is possible to achieve. For this, appropriate "featured games" (e.g., Figure 4.48) can be proposed to them or students can choose games that catch their attention on their own. In addition, students can have a closer look at the code and inspect the programs. Alternatively, the teacher can present showcases that cover a range of games and provide students with an overview on different genres and themes. It is important that the games presented are appealing for all genders (see Section 2.4.2.4), and also show different levels of interaction, but without giving them a specific direction.

5.1.1.2. Game design.

In order to adhere to good game design patterns, it is especially important that teachers allow the students to design their own games from Scratch. The teacher has the option not to restrict the game design process at all, i.e., let them choose their own game elements like story, genre, theme, goal, MDAs and assets (see Section 2.2.1.1), or they may provide a framework (goals or a template), e.g., use of certain properties, design elements, or MDAs. However, teachers must help students in their game design process by explaining the different elements. A promising approach is the use of a storyboard and an agile approach (see Section 2.2.1.2). A storyboard summarizes the different scenes in a game. It holds the "Shape of the Game", and thus considers the core idea, the game concept, the game characters, and their interactivity level (i.e., the control of the characters). At this time, students should also define a meaningful name for the game to create a sense of ownership. Figure 5.2 pictures an example of a storyboard, which is also part of the Appendix (see Appendix A.5).


Name of your game:

1) Title scene	2) Introduction scene
3) Game scene	4) End scene

Figure 5.2.: Template for a Storyboard.

The “Shape of a Game” (see Section 2.2.1.2) has been broadly used by the NOLB students, and especially by girls (see Section 4.4.2). The genres of a game (see Section 2.2.1.1) and the theme provides an idea for the goal and story of the game. Results from the game design inspection in NOLB (see Section 4.4.2) show that female students most frequently used the adventure, puzzle, or jump&run and platform genres, nature and fantasy themes, and the goals of capturing/avoiding or solving. Game design elements consist of game mechanics and dynamics. Aesthetics are to show information for the user to interact with or define the appearance of a game (see Section 2.2.1.1). The MDA encourages the players to succeed. While boys and girls used a similar number of mechanics and dynamics in their games, girls used many more elements for aesthetics (e.g., narrative, fantasy, and sensation). Last of all, students should think about their main and side characters and how to control the objects.

5.1.2. Engaging.

Engagement is an essential part of a playful learning environment. Engagement classrooms should foster collaboration in classroom projects but also allow individual contributions to collections/outcomes. The following compoments reinforce students’ engagement:

5.1.2.1. Warm up.

It is essential to build a safe environment in which students feel engaged by having them ask questions and spark discussion. In that way, students dig deeper and explore more while teachers can correct misconceptions. Many students feel uncomfortable asking questions because they are afraid; thus, respecting and valuing all questions and utilizing active listening must be encouraged by the teacher at the beginning. As described in Section 2.4.1.4, female learners tend to be less comfortable than their male colleagues when asking questions in class (especially if their male colleagues already have experience), and thus are left with lingering doubts regarding the material. Teachers can help students with small group discussions, or address them directly or in small groups.

In addition, many students have an unrealistic perception of technical careers (see Section 2.4.1.1). Therefore, it is important to provide students with more information about "real life" jobs in tech. This provides them with a more realistic picture of their future jobs and explains career-relevant abilities. Students should understand that it is very likely that they need STEM knowledge for their future careers (hardly any job can be done without involving IT or technology) and that there exist alternative routes to IT (e.g., through design, analytics, or problem-solving). This is important to foster their self-efficacy. It is a great opportunity for students if they have the possibility to visit companies, or the option to invite inspiring role models from the industry or universities. Female students in particular recognize the absence of female role models in tech with whom they can personally identify. Thus, female STEM teachers should serve as the most effective role models. Teachers have to explain why computer science is important and its relationship to critical and computational thinking, or problem solving. All of this should reinforce students and should show that people (women) in these areas love what they do, are deeply interested, have fun, and are successful in their domain. In Sections 2.4.1.2 and 2.4.2, examples are provided about how this can be achieved by teachers. In Section 2.4.1.1, two expert interviews showed divergent opinions; they disagree about whether, the gender aspects should be highlighted in class or not. In the right context, the teachers can address this issue and they can ask why the students think that there are fewer women in the IT sector and why. Summarized reasons could be discussed and disproved if necessary.

5.1.2.2. Collaboration.

To foster and allow collaboration, the group constellation and arrangement must play an important role. However, the group constellation also depends on the learning goal (e.g., use of templates) and the course setting. In Section 3.4.1, three types of group works have been discussed: working in small groups (2-5), pair work, or working individually. To foster their sense as a community it is important that there is room for collaboration and communication, even if students work individually. This is provided, e.g. if whole class has the same (learning) goal or sub-goals, etc. Thus, they can discuss and communicate about similar problems and help each other by working all on similar tasks. Observations during NOLB showed that girls formed these kinds of groups more naturally and also split the work more efficiently between different devices than their male colleagues. The gender composition of the team is important as well. During NOLB, students formed teams on their own. Although the team/teachers did not asked for homogeneous teams, only 11 programs were created in mixed gender teams (see Section 4.4.2). If teams are planned, the author proposes that girls should work together in teams. The reasons why are that girls often confirm stereotype expectations in heterogeneous groups and they are more likely to share problems with female partners (see Section 2.4.2.2). During NOLB, different behaviors in heterogeneous groups have been observed. For instance, girls did all the work, whereas the boys did not take an active part in the activity, or boys did the coding and girls did the artwork. To avoid this, a switching of the roles and responsibilities within the project can be a fixed part of the agenda. To introduce students to coding, collaboration, and tinkering is once again the answer, e.g., through-unplugged coding activities and tinkering (see Section 2.1.4.2), and repetitions/discussions/questions rounds on a regular basis to create mutual understanding. During the coding units, if students work in teams they need to be motivated the right way (see Section 2.4.1.2), e.g., in enabling them to assume different identities and different roles or responsibilities within the team (see evaluation in Section 5.2.2). During the coding process, it is essential to provide recognition of their work done, to consider the failure as part of the learning process (celebrate "Aha!"-effects), and to balance extrinsic and intrinsic motivators. In addition, a fun environment should be encouraged in order to foster feelings of enjoyment, which has a positive effect on students' motivation and outcome.

Gender-sensitive education is a key factor for engagement and collaboration as well as the key element of the PECC model. For example, it is important to praise the right way (see Section 2.4.1.2) and to provide a stress-, anxiety-, and competitive-free working environment by considering different skill levels or preferences. Finally, it is important to provide learning materials, example programs, assets, tasks etc. that are free of gender stereotypes and to use gender sensitive language (see Section 2.4.1.2). The coding environment should be something females are happy to study and work in. The effect of enjoyment, happiness, and anxiety increases or decreases students' intentions to participate on similar creative development activities (see Section 2.4.1.4).

At the end of the units, the Constructionism theory (see Section 2.1.3.1) suggests to share and present results in public (e.g., to peers, teachers, parents) to engage the learner, to cultivate their sense of ownership, and to provide students the opportunity for the sharing of artefacts (i.e., through a public forum).

5.1.3. Creativity.

Creativity is born from constraints but can be also killed by them, e.g., through stressful learning environments, or learning goals that do not fit to students' preferences. To support a creative learning, the following tasks are important to meet the "Creativity" aspects:

5.1.3.1. Structure of design units.

The amount of units needed for designing objects, backgrounds, and other game elements depends on the approach used. It is proposed that students have a lot of freedom in creating their games, but with a predefined template or a (learning) goal, certain game elements can already be defined (e.g., the general theme, or the goal of the game). In NOLB courses, students, and especially girls, thought the additional designing units (e.g., during arts lessons) were of great benefit. The design units also depend on how the assets should be produced, e.g., by being drawn by themselves, or using predefined graphics from the internet or a media library. The author proposes to offer students the choice to create their own artwork (either by hand or with programs), but also to be allowed to use graphics from other sources, e.g., to customize and edit them. During the Alice Game Jam event or the Galaxy Game Jam event (see Section 4.5), many assets were prepared which were tailored to these topics and used a lot in the games. It is fun to use sophisticated backgrounds, characters, and items to make the game look good. Teenagers love to play games that are narrative and fascinating, have nice gaming worlds, and boast great graphics and visual designs (see Section 2.4.2.4 and Chapter 4). Thus, they also like to create good games. The teacher's role is to support them in good game design and nurture a sense of pride and ownership among them.

5.1.3.2. Freedom of choice.

Creativity is also constrained by time limits and design restrictions. Students need time to organize their programs, search/draw suitable assets, and to be creative at all. Thus, even if teachers use a template, it should allow customization and personalization (see Section 3.4.2.2). If the design tasks allow room for exploration, customization, and creativity at the same time, students feel engaged (see Section 2.4.2.2). Here again, the question is how assets should be created. The literature supports the argument that female students spend more time on visual customization while boys spent more time on solving logical puzzles (see Section 2.4.1.4). Thus, boys focus more on task completion whereas girls want to create "nice" programs where they are allowed to express themselves to feel a sense of ownership and pride. This could be observed during NOLB especially in students from lower grades.

To conclude, teachers can eliminate the prejudice that STEM professions are not creative. In addition, students can be encouraged to use sound in their programs as well to show their originality. Sounds have rarely been used during NOLB but are also important for producing "good games". However, sounds are often difficult to use in all settings because they can be noisy. Students may have the option to either switch the media sounds on their device to minimal or use headphones instead.

5.1.4. Coding.

The coding component is one of the most important in the field since the course is indeed to teach students something meaningful about coding. Thus, certain issues need to be planned and defined beforehand, as well as planned during the units to keep students motivated. To help teachers with these tasks, the following ideas need consideration.

5.1.4.1. Structure of coding units.

The structure of the course has a big impact on students' motivation (Nah et al., 2014) (see Section 2.1.3.2. First of all, the target group has to be defined, e.g., their gender, age (proposed 12-14 years old, see Section 2.4.1), and the environment (in class, outside school, etc.) Next, the amount of available units has to be set. In Section 3.5.2, performed coding units during NOLB have been presented. The following structure has proven to be successful; Stage 1: Introduction to programming (the Starter); Stage 2+3: the Main Learning (Story and game design and the Coding); and Stage 4: The Closing (presenting/sharing results and assessment). The amount of units in each stage and the overall structure of the course depends a lot on the approached used (A1-A3, see Section 3.5.1) and the focus of the activity. For instance, students who use templates do not need many design units. Since the PECC model is applicable to any coding activity and subject, first considerations concerning the tool used, the (learning) goal that has to fit also to the subject, and the assessment process has to be made. Thus, this task refers again to the main questions of the UDL (see Section 2.1.3.2): the *what*, the *how*, and the *why* of learning. Second, a suitable tool for coding as well as for designing must be defined by the teacher. Learning about coding with the use of visual programming languages is a promising approach and very popular in many primary and secondary schools (see Section 3.2.3). The PECC model can be applied to any programming platform, which supports game development, but is indeed for programming beginners.

To find the appropriate (learning) goal is essential in any setting. A (learning) goal provides the learners with a direction, and smaller sub-goals for different units can scaffold the learning process.

Typically, a learning goal consists of three parts: action, content, and condition (see Section 4.4.3.1). Learning goals should provide a frame for the learner but also allow multiple ways to achieve the final goals and should create a connection to a real world problem to present a challenge (i.e., subject related). This goal should be problem-oriented, and it should support critical thinking (see Section 2.1.3 and Section 2.4.2.1). In addition, it is important that the learning goal(s) not offer a concrete solution in order to stimulate ideas and allow extensions, changes, optimization, and personalization (see Section 2.4.2.3).

5.1.4.2. Personal experiences.

It is up to the teacher if she or he wants to show the tool/platform used, or if the students should explore the tool on their own (or how to introduce students to the main functionalities). The author suggests to present the tools' UI, show students where they can find help (tutorials, videos, forums, etc.), and to create a program together with the whole group (i.e., on the projector). This was a promising approach during NOLB (see Section 3.5). It is very important to choose the right game for this collaborative coding activity. On the one hand, it should attract all genders (used theme, assets, etc. and not strengthen stereotypes), and on the other hand, it should teach/introduce the main functionalities students need for their own games. Thus, this game has to be adapted to both the subject and learning goal and be suitable for students' age and gender. This game can either be programmed by students themselves (one or two students come to the front of the class and add one step to the program) or by the facilitator with students telling him or her the next step. Teachers can also ask students to first create a small program on their own (e.g., with tutorials, step-to-step guidelines) and add enhancements to the game, e.g., add an animation, add a sound, score etc. Relevant coding vocabulary (see Appendix A.9-Glossary) must therefore be explained first. This can be done either through the collaborative programming activity, a presentation with showcases and example programs, or again in plenum with Q&A, or in small groups. One study (Li and Watson, 2011) concludes that students do not see the link between block and visual-based coding languages and more advanced programming (see Section 2.1.4.1). In other words, they do not know that they actually gain coding skills. Thus, it is hard to build interest and confidence. Therefore, this connection must be provided by the teachers.

At any stage, it is important to ask for clarification and to make sure that the group understands the overall concepts. It is also critical to administer regular repetition cycles while asking related questions. To work through the Main Learning or the coding process more efficiently, students can start with a pseudocode or "bricks on paper" activity (see A3). In that way, they think about the needed functions, properties, objects, and their interactions without actually starting to code. At this point, students have to think about how to control their characters, e.g., by using sensors, keys, buttons, physical properties, or animation. Therefore, students need the information about different interactivity possibilities. Otherwise, they will use the most obvious (e.g., Pocket Code's inclination sensor or "When tapped"-condition). An appropriate level of self-directed learning engages students in the learning process and lets them build their own structures of knowledge. While accomplishing small steps, students gain confidence and get closer to the finished program. By breaking down the content into sub-goals (e.g., at the beginning of every lesson) the coding process can be guided and scaffolded. Students need enough time for coding and confirmation if they are on the right track (especially girls, to increase their expectation for success) and subjective task value. For students who are faster, extra tasks could be provided (see Physical Simulation template).

A part of the closing unit is either a presentation of the games (voluntary/mandatory) or a short quiz at the end of the sessions. During NOLB, presentations were often held because most of the teachers did not want to grade their students, but asked them questions like how they divide their work, if they finalized their first ideas or if they had to adapt them, and other experiences. It is a good idea again to encourage all students to ask questions. Subsequently, the games can be analyzed according their functional elements (e.g., achievement of the learning goal, program structure, statistics and the program itself (see Section 4.4.2) or according to the use of game design elements (e.g., use of MDA, level of control, visual design (see Section 4.4.1). Therefore, an assessment sheet has been added to the Appendix (see Appendix A.6).

These results — the PECC model and the PECC guidelines — show not only the tasks that are gender-sensitive but also point out clear actions that are necessary to reinforce (female) students in coding activities (but it is not restricted to girls-only activities).However, the gender sensitive aspects are highly important and considered especially in the following PECC tasks:

- How students are allowed to create their assets and visual design (creativity; see Section 2.4.2.3)
- The group constellations (engagement; see Section 2.4.2.2)
- Promote women in IT, role models and female mentors (engagement; see Section 2.3 and Section 2.4.1.3)
- The approach used, e.g., frameworks and templates (coding; see Section 2.2.2.2 and Section 2.4.2.1)
- Create a safe environment where students are not afraid to ask questions and to strengthen their sense of belonging, and self-efficacy in CS through, e.g., role models (engagement, see Section 2.4.1.2)
- To understand students' different playing behavior, and how to reinforce girls by creating their own games (e.g., if they do not play games at all; playing; see Section 2.4.2.4)
- Development of appropriate example programs, showcases, videos, etc. — the presentation of the tools and provision of a suitable starting point with new a new tool (coding, see Section 2.4.1.5 and Section 2.4.2.5)
- Preparation of material free of stereotypes and use of gender sensitive language in speaking and materials (engagement, see Section 2.4.1.2)
- Let students play games that reflects their interests and are suitable for them (playing, see Section 2.4.2.4)
- During the Game design process: freedom of choice and ownership (creativity/coding, see Section 2.4.2.3) in choosing their genre, theme, goal, MDA, etc. (see Section 2.2.1.1)
- Support of students with storyboards, unplugged coding, and tinkering activities (engagement, coding, see Section 2.1.4.1), and foster their understanding by e.g., explaining important coding vocabulary (see Appendix A.9).
- Foster collaboration and self-expression to build confidence (engagement, see Section 2.4.2.2)
- Gender awareness by ensuring a competition-free environment and observations, e.g., what discourages students in the PECC courses (engagement, see Section 2.4.1)

In the next sections, two case studies are examined using the PECC model. The case studies show how the model can be applied in girls-only situations and how it can serve as a reference for customizing coding tools.

5.2. RQ2/Case Study: PECC Activities for Female Teenagers

Observations during NOLB and the literature review (see Section 2.4.2.5) showed that it is important to encourage female students explicitly to join coding activities. If the pilot schools offered Pocket Code units on voluntary basis in higher grades for all students (because it is only mandatory for student in ninth grade), the majority of participants consisted of boys (e.g., see course 21: boys only!). Section 2.4.1.5 and (Fisher et al., 2015) point out the importance of investigating gender differences in tinkering CS behavior as a factor that contributes to the gender gap in CS. These findings during NOLB supported the argument of using gamified coding activities for girls. The findings were mostly consistent with related studies and suggest the optimal learning environment for girls to be project and game-based and to allow a certain amount of self-expression, collaboration, and creativity by using a range of strategies to address female underrepresentation. With the new developed PECC model and the PECC guidelines the author started to generate a lesson plan (which could serve as an example) for a girls-only course with is part of the Appendix (see Appendix A.7). Thus, the focus of this case study was to examine PECC activities between homogeneous and heterogeneous groups by considering gender sensitive aspects as defined in the PECC model. The aim was to examine if girls felt more engaged in girls-only environments. Although the sample was very small, the insights are very interesting and suggest further studies in the same or different contexts (outside school).

In this section, the PECC model has been evaluated for two months in two classes. The results are presented below to assess how the model can be applied to especially reinforce female students in PECC activities. The aim of the study was to evaluate the impact of the course on increasing interest, self-efficacy, sense of belonging and fun, by engaging girls in design activities, promoting coding and design skills, and enhancing their perception of tech fields. In addition, this section should provide a overall picture or an example how to apply the PECC model successful (step-by-step).

5.2.1. Setting of the PECC activity for girls only.

For the evaluation of the PECC model, an experimental design has been chosen: One class consists of girls only (test group, number of female students = 10) and one class was a mixed gender class (control group, number of male students = 5, number of female students = 7). Students aged 14-15 working in PECC activities with Pocket Code in three double units. Four types of data were collected: 1) observations of students' performance and questions during class, 2) surveys conducted at the different periods during the course sessions, 3) semi-structured interviews with two students of each group, and 4) analysis of projects and assignments completed by the students using the PECC assessment template (see Appendix A.6).

In reference to RQ2 the authors' method should validate the effectiveness of 1) intrinsic motivators (interest, self-efficacy, sense of belonging, and fun) during PECC activities using questionnaires, 2) extrinsic motivators (games, inclusion, framework, goal) via interviews, 3) differences between the groups, e.g., assessment of the programs, class climate, and group constellations via observations, and 4) teacher interventions, e.g., differences in questions and needed guidance between the groups. Figure 5.3 illustrates some impressions of the units.

Figure 5.3.: PECC activities with Pocket Code.

The lesson plan of these units is part of the Appendix (see Appendix A.7). In short, students had three double units with the following structure (see Figure 5.3): Introduction (1 unit) — Design (2 units), Coding (2 units) — Closing (1 unit). During the first unit, a warm-up session with a discussion/questions round about technical topics and games was conducted (see Appendix A.4, PECC guidelines-Introduction). The author served as a role model/expert and explained the student's technical professions, tasks, studies, her career path, and her daily work routine. This happens in a question/answer discussion in plenum. After that, a live-demonstration of Pocket Code was made and a similar program to that one the class was going to program was developed with the whole class. Thus, a powerpoint presentation was used to provide detailed information through animations at some points (e.g., loops, broadcast messages, etc., see PECC guideline-Introduction).

Students should have worked individually but all with the same learning goals. The story/task for their game was "Your character wants to learn something about the planets and goes on a journey to outer space. The planets tell him/her some facts about themselves." Some facts about planets were discussed in plenum as well (e.g., how many planets do you know, name them, is the sun a planet, etc.). The design/coding units included an initial "bricks on paper" activity (see PECC guidelines-Coding); students started to create a poster template with all needed scripts/bricks. Students could draw their assets on their own, use assets from the internet, or lift them from the media library (see PECC guidelines-Game Design).

For the storyboard, each student should have chosen (printed out on paper):

- 1 background (suggested pictures from the media library)
- 1 character (suggested characters: penguin, dog, bird, owl, girl, boy, alien, astronaut, plane; not as a picture but simple the word)
- 3 planets
- 1 set of bricks (not all bricks where described in plenum to foster self-directed learning, e.g., brick “go to front”, printable documents can be found here[92])
- 1 coordinate system (see Appendix A.8, students should first set their character’s and planets’ positions on the paper by defining the a x/y coordinate and think about the size of their objects)

For the first two categories, blank sheets of paper were also available for those who wanted to choose other characters or if they want to draw their own pictures. In addition, a kind of interactive level for the main character was allocated randomly (touch position, i.e. following the finger x/y coordinate or controlling the object by using the inclination sensor of the device). The reason for different interactions was that students should have formed groups to their allocated sensor. The story, the example programs, and the tinkering activity were constructed to be free from gender stereotypes and focused to attract female students (e.g., provide a variety of characters etc.). The learning goals were defined beforehand as follows:

Match of academic objectives:

- The character should be controlled either via touch-property or inclination (allocated randomly).
- Each planet tells you three facts if the character touches it (using broadcast messages).
- Extra task: Add a quiz at the end.
- Use first the set of bricks (on paper) and create a template.

Match of gaming objectives:

- Create an adventure game with the goal of knowledge acquisition.
- Use a background, one main character, and 3 planets in your game (either own drawings, media library, or internet graphics).
- Use the “Shape of a Game”.

[92]Brick documentation: https://wiki.catrob.at/index.php?title=Brick$_{Documentation}$

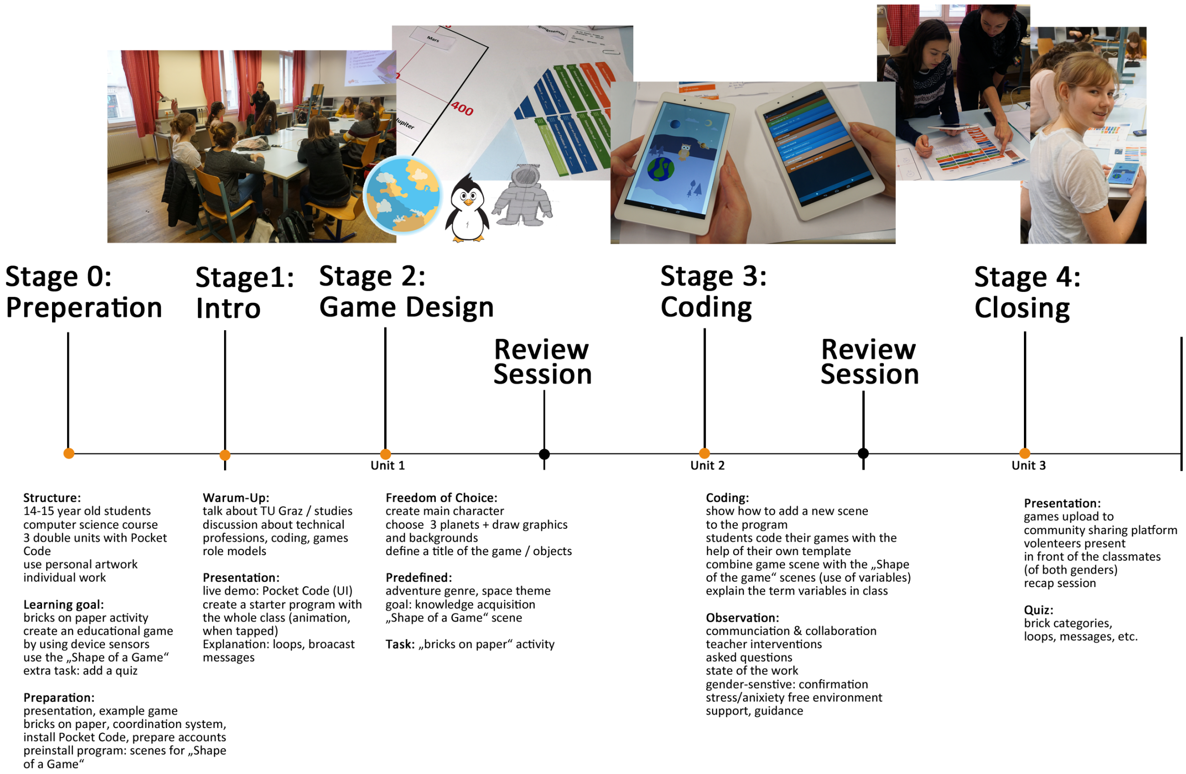


Figure 5.4.: Timetable of the course in reference to the PECC model.

Figure 5.4 also shows two review sessions. These were held at the beginning of each unit with the aim to repeat and focus. In the last unit, a short quiz was performed and two (in control group: one girl/one boy) presented the game to their peers (see PECC guidelines). The programs were not assessed by the teacher. Both quantitative and qualitative feedback was collected from students during the case study, e.g., via surveys, interviews, and on-side observations. The author used questionnaires for asking students about their intrinsic motivators. The surveys also included open-ended questions and a final quiz.

5.2.2. Results on-site observations.

Yelland (Yelland, 1995) defined 17 different categories of students' interactions during classroom discussions. The author adapted these into three kinds of questions: 1) asking for information/explanation, 2) asking for a proposal, and 3) comments (i.e., rhetorical questions). The team tried to provide helpful hints rather them providing them with a closed answer. In both groups, most of the questions were of type number one:

Examples from the control group:

- “For what do I need the brick “go to front”?”
- “Where do I find the broadcast messages bricks?”
- “Where can I enter the speech-bubble text?”

Examples from the test group:

- “How should I start?”
- “How do I add an object?”
- “What can I do that my object do not rotate if it is on edge?”

A question from type two was for example (test group): “Do I have to define the position of the object exactly?”.

Observations of the first units

Observations of the control group showed that the group was clearly separated by gender. The whole group sat on one big table but the girls and boys sat on opposite halves. The boys had different interaction levels but wanted to work together anyway. Students from the test group ignored bricks, which has not been explained in plenum totally, e.g., “go to front”, “if on edge bounce”, whereas students from the control group asked where/why the brick was needed in their storyboard. Two of the girls in the control group cut the bricks very accurately, which took a lot of time. Two from the test group wanted to set the position of the objects very precisely (they even used a calculator). The test group asked less for help compared to the control group but some of them were waiting for input and others asked for every step. Girls in the test group worked in a very concentrated manner and communicated a lot (but mostly about not content related topics). All in all, the climate in the test group was much more pleasant and not as stressful as that in the control group.

Observations of the second units

In the second units, students of the control group did not take the 10 minutes break between the units and wanted to continue with the coding. The girls in that group were faster than the boys were, and had already started to code within the first unit of the double units. All boys started during the second unit. The girls helped each other and one was very fast in adding the needed sensor variables. In both groups, students started mostly with just adding brick by brick. Thus, they made some mistakes and wondered why it was not working at the end. For example, they typed the variables (e.g., sensors) and properties (e.g., “touches_object” property) as a string/text, thus, it was not interpreted by the program (this happened to three in the control group: one boy and two girls, to five in the test group). Therefore, the team told them to integrate smaller steps: for instance, first set the position/size of the object, second, add the control of the object, and third, handle the interaction between the objects. Most importantly, test the game after every step. This helped them a lot. One boy said he finished his program and was not motivated to work further on it. The next unit shows that he did not fulfil the learning goal and he had some mistakes in the scripts. The boys in the control group fooled around and the whole classroom climate was very noisy. Only one of the boys worked in a concentrated way on his program and one other boy drew his character on his own (a bird). Again, girls from the test group asked a lot and almost in every step. Since the team was curious about the questions asked, collaboration was not fostered explicitly (as recommend in PECC guidelines). Students preferred to ask the facilitator and did not help each other that much. At the same time, they were very interested in the units, in coding, and the results. Anyway, the team often had the feeling that they were more task oriented than intrinsically motivated. One girl was very frustrated because she had the feeling she did everything wrong and she was very insecure in each step.

Observations of the third units

This time the boys in the control group collaborated a lot and were more concentrated in finishing their games. At the end of the unit, one boy presented his game voluntarily to his peers. Students in the test group were busy making their programs more perfect. For instance, they asked how to make the object not flip (if using the inclination sensor and "if on edge bounce" -brick). They also asked why the object which should have followed the touch property appeared at the beginning in the middle of the screen although they had set the position somewhere else (because "finger_x/y" is at the beginning zero). Two girls from this test group were very silent and shy. For example, they waited instead of asking, or ignored obvious mistakes (e.g., the object was not moving even if she used the inclination sensor correctly but she needed to multiply it to get a larger radius). One girl was very fast and finished 10 minutes earlier than them. She helped the two other girls in the group with including the end scene (use of variables) as the team asked her to do so.

5.2.3. Questionnaire results.

The questionnaires (pre, daily, and post) handed out to students included measures of the various factors regarding students' intrinsic motivators used in the PECC model: interest, sense of belonging, self-efficacy, and fun (see Section 2.4.1). The pre-questionnaires aimed to collect students' perceptions about the course, coding, and technical fields. The daily questionnaires (handed out at the end of the first and the second units) had specific questions to the unit covering daily interest, fun and achievement. Finally, a post-questionnaire asked questions about the PECC activity itself. Table 5.4 lists the questions defined as an indicator for each of the components. This questionnaire is considered as an important outcome of RQ2 because it will be used also for the evaluation of future PECC activities to predict intrinsic motivation in (female) students.

Table 5.4.: The indicators used in the survey (pre, post, and daily).

Construct	Pre-questionnaire	Daily questionnaire	Post-questionnaire
Interest	I'm interested in learning something about coding.	The coding unit today was very interesting.	The coding course was very interesting.
	CS/coding is boring.		The coding course was boring.
			I would have liked to spend more time with coding.
Sel-efficacy	I think I will like coding.	Today I achieved everything that I planned.	I liked the coding.
	I generally feel confident in using my smartphone/ computer. I have no problem in math.		I felt confident during the coding course.
	I have less self-confidence in CS.		I often felt helpless.
	I think I can learn coding.	I learned something new today.	I learned something about coding.
		I'm proud of what I achieved today.	I'm proud of my final program.
Sense of belonging	I can imagine what people do in technical professions.		I got many ideas what people in IT do.
	I can imagine what programming is about.		I now have a better understanding of what coding is about.
	I think a job in coding would suit me.		I think I have the same/similar characteristics as people who are working as programmers. Programming reflects my thinking.
	I can imagine that programming is important for my future job.		I think this new knowledge about coding is useful for my feature.
		I was able to work with others today.	I felt engaged.
		I worked mostly alone today.	
Fun	I'm looking forward to the coding units with Pocket Code.	I'm looking forward to the next coding unit.	I would like to participate in similar courses.
		The programming unit was fun for me today.	The programming unit was fun for me. I found the units stressful.

In all cases, a 4-point Likert scale was used to measure the variables (Sullivan and Artino, 2013). In addition, two questions were added to ask about their intention to use Pocket Code further or to recommend it to friends (post-survey). Additionally, in the post-questionnaire it was further examined who they asked for help (classmates, teacher, tried it alone). The questions have been developed at the basis of literature by Schwarzer Jerusalem (Schwarzer and Jerusalem, 1995), the CATS Attitude Scale Items (Krieger et al., 2015), the GERD model (Gender Extented Research Development) (Claude et al., 2014; Giannakos et al., 2014), and from other researches (Li and Watson, 2011). No questions have been asked that could foster stereotype threats, as proposed (Krieger et al., 2015), e.g., "Girls can do technology as well as boys".

The same size of this quantitative evaluation is very small (number = 23) but should provide first insights. In the future, the author is planning to use the PECC model for more courses as well as the surveys to provide results that are more significant. The average results are visualized in Figure 5.5 — 5.10 separated by group. The "4 Likert Scale" refers to: 1: strongly disagree, 2: disagree, 3: agree and 4: strongly agree. It is recommended to use no neutral value and to use counter questions, e.g., Coding is interesting — Coding is boring, to demand their attention (McLeod, 2008). Thus, it is not always "the higher, the better". Question with "the lower, the better" are marked with a "*".

Therefore, four hypotheses have been defined in validate the intrinsic motivators in the PECC model:

- H1: Female students are more interested in the coding in gender homogeneous classrooms
- H2: Homogeneous classrooms provide girls with more self-efficacy in performing coding activities
- H3: Homogeneous classrooms strengthen girls in their sense-of belonging
- H4: Female students have more fun in homogeneous coding classrooms

The results will be discussed in Section 5.4.2. At the beginning, both groups were equally interested in starting with the coding lessons (see Figure 5.5). Students in the test group showed an increased interest in all double units with an average of 3.3 at the end. Compare this to the control group, which was after the first unit, less interested (average 2.70) and at the end back to the same level. Most of the students from the test group disagreed that coding is boring and most stayed with their opinion after the course (average 2). In the control group, in the final unit, most strongly disagreed that coding is something boring (average 1.58). Both groups agreed that they had wished to spend more time with the coding part.

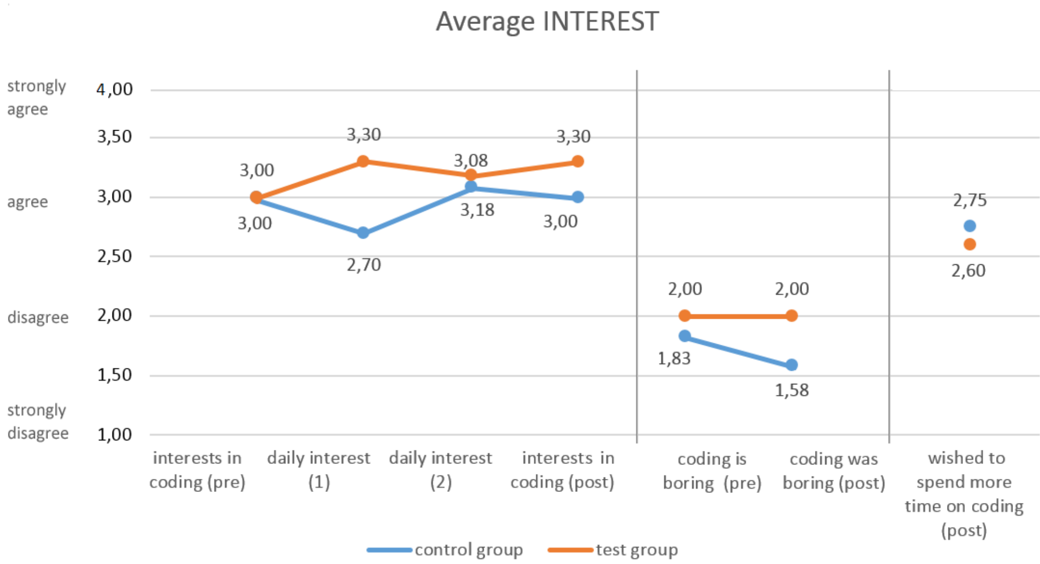


Figure 5.5.: Student interest in the coding units.

Both groups agreed/strongly agreed that they thought they will like coding (average 3) beforehand, but the control group said it was more likely that they did not like coding after the course (average 2.58), whereas the test group only slightly changed their opinion (average 2.80; see Figure 5.6). The same can be observed in their confidence level: Students in the control group started with a higher perception of their confidence level (3.04), but they did not feel that confident in actually using the app (average 2.00). Students from the test group again had only a slight differences between their perception and reality (difference 0.10). Here, the counter question confirms the assumption (no feeling of helpless). On the one hand, students from the test group agreed less that they could learn how to code (average 3) but agreed more often that they learned something new after the first (average 3.80), second (average 3.45) and last double unit (average 3.30) than students from the control group. The highest difference can be found in the questions where the author asked if they felt proud of their achievements. The control group mostly disagreed (average 2.10) whereas the test group mostly agreed/strongly agreed (average 3.4). During first double units, students did game design and the "bricks on paper" activity. Students of both groups were again similarly proud of their final products (average of both 2.86).

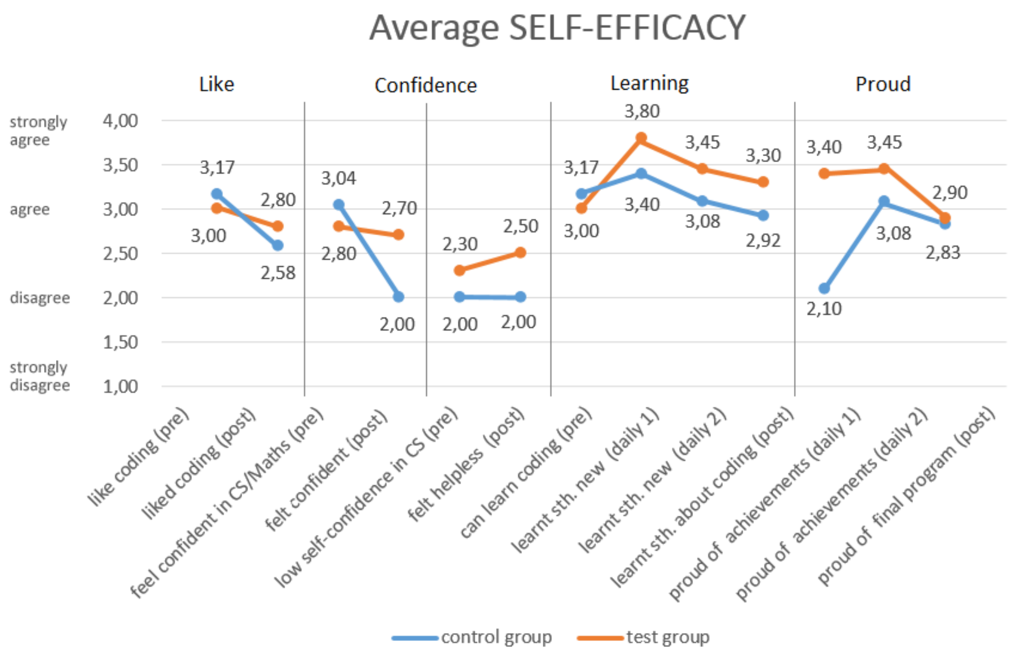


Figure 5.6.: Students preferences, confidence and satisfaction.

Both groups started with less knowledge about IT professions and coding and agreed that they had a better knowledge of both after the course (see Figure 5.7). Students from the test group mostly strongly agreed that they now knew something about coding (average 3.90). All students disagreed more often that a job in the IT section would suit to them, but students from the test group showed a slight increase (difference 0.2) after the last coding unit and students from the control group showed a slight decrease (difference 0.13). Students in both groups agreed slightly more often after the last unit that coding will be important for their future (average 2.21). A closer look at their engagement level showed that students from the test group collaborated more in the first unit (3.80) than in the second (average 3.17), and students from the control group worked alone more often during the first units (average 2.30) than in the second units (average 3.80). This is in line with the onsite observations. Students from both groups showed a similar engagement level (average 3.24).

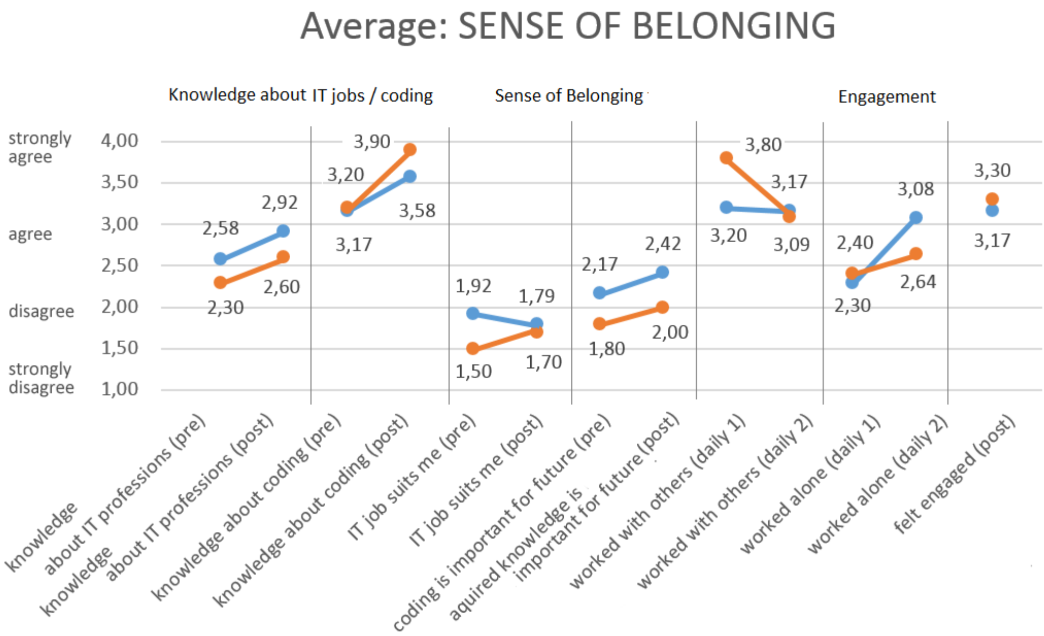


Figure 5.7.: Students' sense of belonging and engagement level.

Students of both groups started with a very similar level of anticipation (average 3.22, see Figure 5.8). Students from the test group after the first units agreed more often that they were looking forward to the next units (average 3.80) but they more often disagreed that they want to join similar coding courses at the end (average 2.1) than the control group (average 2.58). Students from the test group had a consistent fun level (3.50) but disagreed more often after the last unit that they had fun (2.80). Overall, students from the test level had a higher fun level than students from the control group. Students in both groups disagreed that they felt stressed during the units.

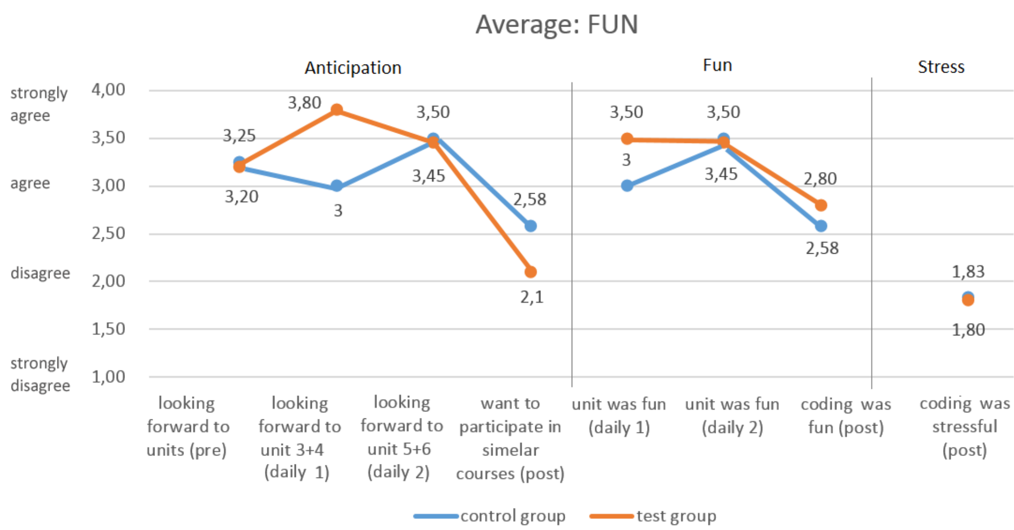


Figure 5.8.: Students' fun level.

In the post-questionnaire the team asked two more questions regarding the intention to use Pocket Code in the future (see Figure 5.9). students in the test group agreed more often that they would tell their friends about the app.

In addition, the team was asked who they asked for help during the units. Whereas in the control group all three options were mentioned (1.) I tried to find out by myself, 2.) I asked a classmate, 3.) I asked the teacher), students from the test group asked their classmates or the teacher more often than to try it out on their own.

Within the daily-questionnaires the team additionally asked two open questions:

- What did you like the most?
- What did you like the least? Recommendations?

The result is illustrated in Figure 5.10. Note that some answers fit different categories (e.g., I liked the coding and to create my own characters) and that every student filled out this questionnaire two times. First, the positive comments are described. Most answers can be summarized in the categories of "working process" and "coding related answers". Within the category, "working process", answers mentioned by students in the control group were the design aspects (6), e.g., to design characters, or they mentioned the "bricks on paper" activity, freedom of choice aspect, or the pleasant classroom atmosphere (it was not too noisy). Within the category, "coding aspects", answers mentioned by students in the control group covered the coding itself (3) and the result (2), e.g., the progress and seeing what happened in the code. Two students mentioned as an "organizational" aspect that the power point presentation that was used for explaining the theory was appealing.

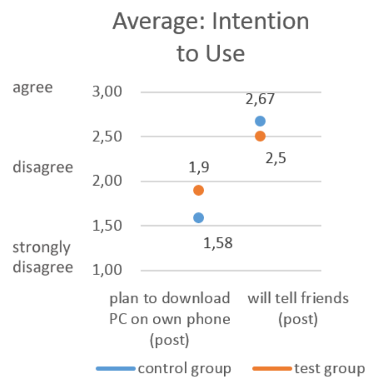
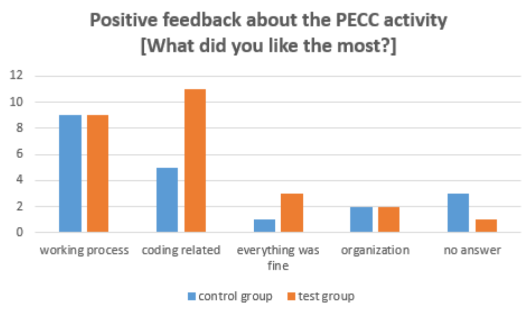


Figure 5.9.: Intention to use Pocket Code in the future and if they tell their friends about the app.

In contrast, students in the test group had many more answers summarized in the category "coding aspects". They mentioned the coding itself (5), the result (2), e.g., that "I finished the program", and also the learning aspect (4), e.g., that "I learned something about coding". Within the category "working process", the answers covered freedom of choice (2), working in groups (3), or working independently on the program (3). Two students mentioned as an "organizational" aspect that everything was explained well and three said that everything was ok.

Negative aspects that have been mentioned by the control group showed that they wished for a more precise explanation at the beginning and more group work. Most of them left this field blank. One student mentioned that he did not like to prepare the facts about the planet and two criticized the questionnaire (no neutral option). Students in the test group mentioned 11 times that everything was fine or perfect. Two students also mentioned that they preferred to have more explanations at the beginning and one said she wanted to have more freedom in designing the game.



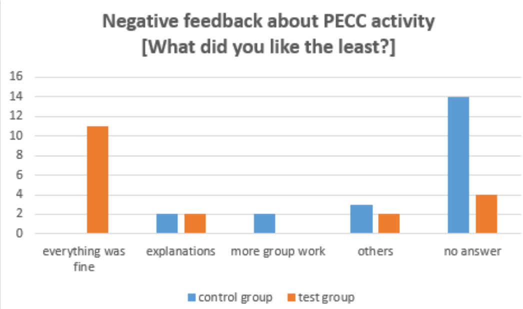


Figure 5.10.: Analysis of the open questions regarding likes and dislikes to the coding units. Students answered two times (after double unit 1 and 2).

5.2.4. Interview results.

The author conducted two interviews with female students (two from each group) one week after the last unit. The goal was to get further information about the working process, their coding experiences, recommendations, and their impressions to validate the extrinsic motivators set in the PECC model. Since the number of participants in this interview was small, special comments are listed below (Creswell, 2009; de Hoyos and Barnes, 2012).

The following questions have been asked per group:

1. How did you like the coding course?
2. How was your experience in the homogeneous/heterogeneous group?
3. How would you describe the group dynamics in your class?
4. Did you like to have an individual program or would you rather prefer to work more collaboratively?
5. How did you like the “bricks on paper” activity?
6. How did you like the idea of the game (planets, space)?
7. Would you like to have had more freedom in the game design?
8. Additional comments to the course . . .

Table 5.5 summarizes the main ideas per questions and interesting quotes. They were either repeated by the interviewee, surprising/interesting, or information that seemed to be important for themselves.

Table 5.5.: Results of the interview following the PECC activity for girls-only

Q#	control group	test group
1	“I am not the kind of person who does coding. After the course, my opinion changed a little bit but I’m not interested in such things at all.” “It was funny but also complicated, for example where to find the different bricks (. . .).”	“Well the app is generally great (. . .) I would not use it in regular basis, you know.” “Certain game functions are missing, that you can start immediately using the app, like predefined building blocks you can switch on.“ “The app should be more colorful. Everything is so gray, blue, and dark.” “You should be able to play more insteadr,e of programming. Pre-prepared objects would be great.”
2	“Boys also asked questions which were interesting for me as well and that I did not think of. Thus, the lesson has been improved by them.” “Everybody worked on his own program and it made no difference with the boys in the class.”	“It was better that we were just girls because a few boys from our class are already familiar with real programming languages and they would just make stupid comments if we ask something.“ “Exactly!”
3	“If we can decide on our own, we usually build groups with the same people. “ “But there is no one in our class I would refuse to be in the group with.”	“(. . .) we can ask everybody in our class because we are all in the same boat.” “For example, in computer science we can simply shout for help if we don’t understand something and someone who has already finished the task will help you.”
4	“I am fine with both but I personally prefer to work in a group because you can ask each other for help.” “Classmates explain it in their own words which makes it sometimes easier to understand.”	“It was ok that everyone created her own program, because I think in the group you have to (...)" “(...) it would be much more chaotic.” “Yes,exactly” “You have to find a compromise. We both once visited a “Legoleak” programming course and we were in the same group and it was so chaotic.“ "Yes! Who should do this and who should do that, and what should we do anyway(. . .)”

5	"(. . .) actually I found it useful that we should cut it out first and then we had our own template." "You could always verify while working on your program." "The lesson was more versatile (. . .) we did not only do the coding stuff (. . .) I think that was better."	"The lesson was much more structured (. . .), but I'm that kind of person who wants to try everything out right away. Well I mean, I think they others benefit a lot from this approach but for me I just wanted to learn programming!" "Initially I had no plan about the app itself and what I should do. I am not that good at memorizing things, for example if somebody explains or shows things to me in a powerpoint (. . .) I used my template very often and I think it helped me in memorizing."
6	"The idea was good. We were allowed to design our own planets and characters and customize them." "Since we did it in school, I think it was good that the goal was school-related."	"(. . .) basically it was not a game that you want to play several times (. . .) it would have been great if one can go to the planets as well." "(. . .) I imagined that the planets are rotating and that I have to jump from one to another planet. That would have been awesome." "Yes, that you learn something by doing something fascinating."
7	"It was good that we had a fixed goal. If I had to think on my own, it would have been much more work and I would have needed more time (. . .)".	"(. . .) it was enough for three double units." "It was good that you gave us the goal. We really knew: Okay, I have to do that and I'm trying to do that now." "I think with this introduction game we are now able to code our own games for example to visit the planets."
8		"(. . .) the units were not stressful and we had enough time." "Learning a real programming language takes a lot of time and is exhausting with Pocket Code you can code within minutes. I think that's great!"

5.2.5. PECC program assessment.

The assessment of the final programs was done with the help of the PECC Assessment template (see Appendix A.6). The programs have been evaluated on the basis of the game design elements used (visual design, main characters) and learning goal achievement (matching of learning and gaming objectives). Since every group used a pre-defined template of the "Shape of the Game", which they could personalize, the amount of scripts, bricks, objects, etc. was very similar throughout the programs.

Game Design: In the test group, three girls drew their planets on their own using Pocket Paint. For their main character, they all used mostly the proposed animals like penguins (6), owls (6), a raccoon (2), fish, birds, or dogs. One girl in the control group used a plane and one girl in the test group used an astronaut which she drew on paper. One boy in the control group drew his bird as well. The background was mostly space themed (13), or winter wonderland themed (5) and all downloaded from the media library. Two used fantasy, nature, or sky as a background. The inspection of the program showed that students from the test group often placed their speech bubbles outside the screen (5) or their background look was too small (4, test group=3). Students in the test group mostly used one speech bubble after the other (7), or emojis within the bubbles (3). They also included their own graphics in the "Shape of the Game" screens (2: astronaut, penguin).

Engagement/Coding: All students from the test group achieved the academic learning goals, whereas 4 students in the control group did not reach goals (one of the control group did not achieve one subgoal: no real facts about the planets, and three students from the control group even failed in 3 subgoals, e.g., there was no interaction between the character and the planets at all, no "When" condition, or no broadcast messages). All gaming goals were achieved by 2 students of the control group and 3 students of the test group. In both groups, most did not integrate the end screen in their games, thus used variables to determine that all planets provided information. This was only done by three students in each group. All students in the test group integrated the title and instruction screen whereas 4 of the control group did not use the "Shape of a Game" at all. Figure 5.11 shows all programs of this PECC activity (left: programs of the control group, right programs of the test group).

Figure 5.11.: Programs and storyboard of the PECC activity.

5.2.6. Results quiz.

In the last unit, a quiz was conducted with the whole group. This quiz consists of five questions:

1. What happens if you tapped on the play button?
2. Each object consists of: (a, b, c)
3. Which brick categories can you remember?
4. Which loops did you use or which ones can you remember?
5. What do you need for objects to communicate with each other?

These questions have been discussed at the beginning of the third lesson (during the review session). The results of the quiz are shown in Table 5.6.

Table 5.6.: Quiz questions and evaluation

Q#	answers	control group	test group
1	the game (scene) starts it displays what you have already programmed you can see mistakes in the code thus you can improve the code	12/12	10/10
2	a) Scripts b) Looks c) Sounds	12/12	2/10
3	Event, Control, Motion, Sound, Looks, Pen, Data	8/12	10/10
4	Forever, repeat until/while/x-times	7/12	4/10
5	Broadcast messages	10/12	9/10

A more precise answer for Q1 would be: the scripts/bricks/code is/are executed. For Q2 most of the students knew about the scripts and looks in an object but wrongly mentioned event or motion which are brick categories. In the test group only two out of ten remembered all three categories. For Q3 three students in the test group mentioned no category at all. Students got it right if they know at least three different categories. Three students from the control group knew all seven categories and two students remembered six categories. From the test group, students were able to list mostly three (5) or four categories (5). Most often they mentioned the categories Event, Look, and Motion. For Q4 most mentioned two kinds of loops. For Q5 nobody wrote messages but they all mentioned the bricks to achieve the desired behavior (e.g., If you receive, send to all).

The results will be discussed in Section 5.4.2. In the next section, the Pocket Code itself has been adopted with the help of the PECC model to be more suitable for female teenagers.

5.3. RQ3/Case Study: PECC to Customize Pocket Code for Female Teenagers

To answer RQ3, first, a focus group discussion was conducted to consider the needs of our focus group at an early stage of development. As a result, important insights for the design of the app were collected while new ideas for feature games were created as well. These games have been subsequently designed by university students from the master's degree program "Industrial Design" at the University of Applied Sciences during a Pocket Code workshop. Second, Pocket Code has been evaluated with the help of the PECC model to examine which customizations of our services and tools were necessary to strategically reinforce female teenagers in PECC activities. The GERD model (Claude et al., 2014; Maaß, 2018) and the findings from the literature (see Section 2.4.2.5) provided again insights to develop this new version. In this section findings for and against have been presented, while sections of 2.4.1 supported the argument that girls have a different technology affinity, as well as different intrinsic and extrinsic motivators like different playing behavior/interests and affections. The concept of the "strategic essentialism" (Spivak, 1990) shows that is sometimes necessary to strengthen a specific stereotype in order to reinforce a group in long term. During NOLB, the team got a clearer picture what kind of games, genre, themes, characters, etc., girls tend to like and play (see Chapter 4). These were obviously not the same gaming preferences as those preferred by boys. Thus, as a practical output of this thesis, a closed beta apk of this new flavor version has been developed and uploaded to Google Play to conduct first user tests.

The Create@School app, which was developed during NOLB (see Section 3.4.1.1), is geared towards schools and teachers. This individualized version of Pocket Code for schools already shows great potential to create even more tailored tools (versions) and services to focus on specific target groups and reinforce them as a result. This stands in contrast to the "One Size fits All"-solution Pocket Code which may discourage several user groups. The author of this thesis is convinced that a "flavored" version especially for girls would increase the number of female users and would thus strengthen our female community as a result.

This section first presents the results of the focus group discussion as well as the game ideas developed by university design students. Subsequently, the concept for the new version for teenage girls will be presented in reference to the PECC model components of Playing, Engagement, Creativity and Coding.

5.3.1. Evaluation group discussions.

To create a concept for the new version for female teenagers, a focus group discussion with 10 girls of two different age groups (12-13 years old, 14-15 years old) was conducted. This focus group discussion was shepherded in October 2017 at "Akademisches Gymnasium". During this interview, girls created their "perfect" game ideas with the help of a playful design activity. The literature suggests (Apptory, 2018; Onwuegbuzie et al., 2009) that the facilitator of such discussions should pay attention, listen, be active, be respectful, keep track of time and context, ask open questions, and ensure that all participants speak and feel comfortable. In contrast to focus group discussions during NOLB (see Section 4.3.4), where the team asked many specific questions, this discussion was more open and creative.

5.3.1.1. Preparation and setting.

For the preparation of the focus group discussion, several resources have been created to efficiently ensure the involvement of the target group in the development process of the new app:

1. A questionnaire for defining a persona of each participant

This questionnaire consists of four parts. In the first part, demographic and questions to interests have been asked (age, hobbies, favorite pet, star, movie, dream job, etc.). In the second part, questions about coding and school subjects have been asked (coding experiences, favorite subjects). The third part consists of questions related to smartphone use (do you have one, which OS, usage, favorite apps, new apps, etc.). In the last part of the questionnaire, questions about the student's playing behavior have been asked (e.g., games, format, genre, favorite games, etc.).

3. Three potential app names for the new version have been defined

The idea was to already have three names for the app and to let the girls talk about the different names and their associations with the names.

For an appropriate app name, certain criteria should be fulfilled. The list below summarizes recommendations from Google posts/blogs and team experiences and seems to be the most important in reference to our target group of students under 15 years:

- The pronunciation should be clear and easy for international speakers.

- It should be clear how to write it if you hear it, which is also necessary for international speakers.
- Check for any bad meanings of similar sounding words in other languages.
- The new name must not be uncool for teenagers (main user group).
- There should be a first impression or "story" behind the name.
- It should sound nice and positive, with a warm feeling to it.
- It should use only two syllables like Google, Yahoo, YouTube, Apple, etc.

In addition, it is important that no existing trademark in the US, the European Union, China, Korea, or Japan exits, and to check for domain names and Google search results. In reference to the list, potential names have been defined by different groups. One app name has been created by the Catrobat design/UX team, one by our focus group (a class of 19 girls at Akademisches Gymnasium which had used Pocket Code already), and the last app name by the author herself. The three names are:

- Pixie Code (Catrobat UX/design team)
- Make(IT)up (focus group)

As a class activity with Teacher 7 (see Section 3.5.2), the girls had the task of collecting groups of 3-4 app name ideas for the new version for girls. This class already had Create@School lessons during NOLB (see course number 19). At the end of the session, seven names were proposed by them. Every girl could vote several times and could use two votes at once for one name. The result showed the following: Make(IT)up (21 votes), Nail.IT (20 votes), PrincessCode (11 votes), [PP] PrincessProgram (11 votes), Girls Clip (3 votes), SpeedGames (1 vote), and GlitterSplitter (1 vote). Unfortunately, not all names have a positive meaning but the girls who have mostly German as a mother tongue were not aware of this.

- Luna&Cat (author)

Our project is inspired by the MIT Scratch project. The Scratch platform has the "Scratch Cat" as a logo (see Figure 5.12). In addition, "Luna Cat" is a side character of the animation TV series "Sailor Moon" which was very popular in the 90s (and also loved by the author) (see Figure 5.12). This was the author's inspiration for that name (the author owned also a cat with the name Luna). With "Luna&Cat" there are no copyright issues and it should refer to a girl and her cat.

Figure 5.12.: MIT Scratch logo (Resnick et al., 2009) and Luna Cat from Sailor Moon (MyAnimeList, 2015).

These three names were printed on paper to provide one starting point for the discussion.

4. Several colored community webshare screenshot examples (see Figure 5.13).

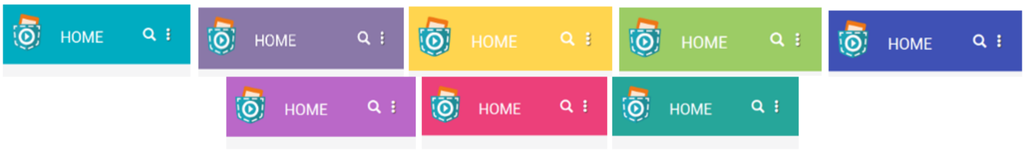


Figure 5.13.: Screenshots that show different colorings of the community webshare platform.

The first screenshot shows the default community webshare color, and the second (violet) was the themed community webshare color during the Alice Game Jam event of the wall of fame page. The goal was again to provide a starting point for a discussion and to see associations to different colors.

5. Idea cards for the game idea development

Since, most girls are not that familiar with playing games (most of the 10 girls conformed to this), the author provided some idea cards with defined genres, themes, and characters:

- Genres: adventure, action, puzzle, quiz, strategy, simulation, platform, shooter, RGP, jump'n'run, Skill, racing, music, education, sports
- Theme: criminal/detective, science fiction, fantasy, time travelling, middle ages, horror, romance, travel, sports, fashion & trends, nature, jungle, desert, space, future
- Characters: animals, witch/magician, zombies, vampires, aliens, monster, mythical creatures, girlfriends, stars, pony, cars, woman, man, werewolf, ghost, king/queen, princess/prince, hero, children

These terms were printed on three different colored papers to help the group in their game design process. First, the girls got blank cards with the task of thinking about a genre, theme, and characters (these terms have been discussed or explained to them beforehand). The goal was not to influence them but first to show them what they could use and second, they received the idea cards as a support.

Both focus group discussions were conducted on the 30th of October 2017. First, with five female students in eighth grade (students aged 13-14) and second, with five female students in seventh grade (students aged 12-13). The duration for both was 50 minutes. To motivate them, one Pocket Code t-shirt was drawn among them at the end of the day. Every student was allocated a number (1-10) to protect their anonymity and their parents had to sign a letter of consent in advance.

First, the context was explained by a brief description of Pocket Code (functionalities, what you can do with it, etc.). Second, the rules and goals was described to the participants:

"The aim of this focus group discussion is to learn how we can improve Pocket Code, so that the app can be optimized to your needs and requirements. That's why I am looking forward to your inputs and opinions. Your feedback will be treated anonymously and passed on to our designers and developers. I ask you to answer honestly. You are the ones who will speak. I want to hear everyone's opinions, so it may be that I will address you directly. There are no right or wrong answers: each of your opinions is important. Everything that is discussed will be recorded on tape. However, you should still be aware of your anonymity and not mention your names. Are there any further questions?"

Third, the schedule was discussed, and then they filled out the questionnaire. After that, as an ice breaker we discussed their favorite pets and stars. Then, the group continued with discussing the app name, community webshare color, and finally, they started thinking about their game ideas with the help of the idea cards. Figure 5.14 provides an impression of the creation process of the game ideas.

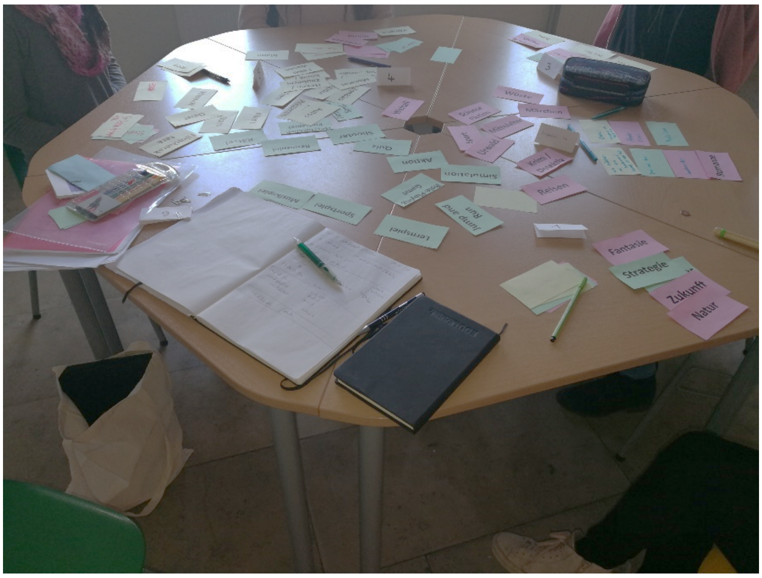


Figure 5.14.: Game ideas creation process during the focus group discussion.

5.3.1.2. Results: focus group discussion.

For the evaluation in short, answers from the focus group to the app name and the community webshare colors will be presented. Furthermore, the game idea per participant will be described. These ideas were subsequently used as an input for a design workshop at the University of Applied Sciences (see Section 5.3.1.3). The goal was to integrate these games as featured games for the new version for girls.

Results: app name and community sharing platform

The evaluation concerning the app name in seventh grade shows the following results:

- Pixie Code (1), Make(IT)up (3), Luna&Cat (1)

The one girl who chose Luna&Cat said that she loves cats and Luna is a nice name. Make(IT)up was chosen because the other two names sounded boring and this name is very appealing, one can imagine what kind of app it is, and they would definitely download it.

The opinions about the app name from eighth grade were the following:

- Pixie Code (1), Make(IT)up (4), Luna&Cat (0)

Pixie Code was described as too similar to Pocket Code or as too childish. Make(IT)up was preferred because it sounds girly and it already gives a clue about the kind of app it. Luna&Cat did not made any sense to them.

The discussion about the coloring for the community sharing platform showed the following in seventh grade: One girl said that she would prefer violet because it is not used only for girls like pink or red is. In addition, they mentioned yellow, green, magenta, or turquoise as fitting. A reason named was that these colors are very bright and friendly. Two of the girls said they love pink.

In eighth grade, the results showed that all said that they would prefer pink or red as their first choice because the app is for girls, even if they did not consider them as their favorite colors. Other mentions were turquoise or green. The color red was associated with happiness as well.

In addition, the facilitator asked the group what the app would need so that they would download it and use it. Participants from seventh grade mentioned that it should have many functionalities, be well-structured, provide updates with new features, and not be too difficult to use. One girl said that she wants to have pre-prepared assets and that it should provide the possibility to customize assets (e.g., to create a man with blue eyes-like in "The Sims" or "Movie Star Planet"). Participants from eighth grade said that the app should be well structured as well, and it should provide tutorials. The app should be mostly self-explanatory (one said she hated to watch tutorial videos). It was also mentioned that the app should not need too much memory or battery and no internet connection should be requested.

Results: game idea interventions

Although 4 out of 10 said in the questionnaire that they never played any computer games at all, all of them knew exactly what their perfect game should look like. In the Table 5.7 (participant no. 1-5: eighth grade) and in Table 5.8 (participant no. 6-10: seventh grade), are the idea cards used per participant and the idea of each game is presented in short.

Table 5.7.: Game idea: participants number 1-5, Grade 8

Participant number 1	
Genre	adventure, strategy, RPG, jump'n'run
Theme	nature, future, fantasy
Character	animals with special skills, people to customize, a witch, fantasy creatures
Other notes	no shooter weapons, open world with quests
Game idea	open world (but limited), you can choose characters by yourself (define their special skills), you have to help people, if you fulfill a task you get something, e.g. tools, or something to wear or to build other items, the game take place in the future: apocalyptic mood (survivals, zombies)
Participant number 2	
Genre	jump'n'run, music
Theme	romance, music
Character	a girl on a skateboard and friends to meet
Other notes	skateboarding, piano in the background
Game idea	a girl on a skateboard, she meets a friend, there is a romance with a boy, you win if they get married (on the skateboard!)
Participant number 3	
Genre	fashion, RPG
Theme	Make-up, hair, clothes, farm, horse stable
Character	old women, a farmer and animals
Game idea	an elderly woman enters a fashion shop, you have to put on makeup until she is young again
Participant number 4	
Genre	RPG
Theme	criminal/detective game
Character	Red John, Patrick Jane, Richard Castle
Other notes	you can choose if you are the murderer or the detective in the game
Game idea	a detective game, but they should be clever not as dump as in the most games, it should be cool like in "Death Note" or "The Mentalist"
Participant number 5	
Genre	RPG
Theme	art, nature
Character	pony, dog (golden retriever), a flower
Game idea	you are a flower that has to pass different tasks, e.g., minigames or jump and run, there is a dog and a pony who give you new tasks, you become friends with them, you travel around the world with friends and keep leveling up.

Table 5.8.: Game idea: participants number 6-10, Grade 7

Participant number 6	
Genre	adventure, jump'n'run
Theme	Disney, forest, future, desert
Character	children, animals
Game idea	the children must save the future of the planet, a machine should bring the world back to normality (if it is switched on), but if the children become older they will become cyborgs, therefore, they have to collect certain magic potions which make them young again, they have to run until they find the machine (there are 50 levels), you have a limited number of time or meters, if you do not collect all the potions, you will become older and turn into a cyborg
Participant number 7	
Genre	adventure, sport
Theme	horror, middle age, nature, jungle
Character	animals (a tiger)
Game idea	the game takes place in the jungle in the Middle Ages, you are a tiger-it is scary and horror-like, the tiger baby has been kidnapped and you have to save it
Participant number 8	
Genre	adventure
Theme	fantasy
Character	witches and magicians, people
Other notes	create new worlds, people who develop, fun
Game idea	create your own characters, travel to different worlds and get better; there are fantasy people, evil and good sides
Participant number 9	
Genre	puzzle, quiz
Theme	space
Character	prince, princess, stars
Other notes	mix of cities, landscapes, animals
Game idea	you play in a small world which is totally blank, you build your own city, you meet people who ask you questions and provide puzzles per level, with every quest you level up; in a higher level you get a royal palace (become a princess and then a queen), you can go to concerts and plays infinitely (no winning-no losing)
Participant number 10	
Genre	action, adventure, strategy
Theme	horror
Character	fantasy creatures, ghost, birds, cats, lions
Other notes	to save somebody, collect items, missions
Game idea	multi and single player mode is possible, you can customize the character by yourself, children are searching for their parents, there are 3 keys in every world to find which one opens a new portal into the next world, there are obstacles, good and evil people, a boss fight at the end of every level, 3 books must be found to open the last portal to find the parents again.

These creative game ideas built the basis for the Pocket Code workshop with master students at the University of Applied Sciences.

5.3.1.3. Results: persona and games.

A number of 14 university students (female students = 9, male students = 5) from the first year of the master's degree program "Industrial Design" took part in this 10 1/2 hour workshop (14 ECTS) during 2 weeks in November 2017. The university students aged 23 to 28 divided into groups of five. Most of them said they do not play games in their leisure time (only three of the male students play games and one of the female students). The goal of the workshop was to create or design one of the 10 game ideas invented by the focus group (see last section). Therefore, they got the idea cards of one participant (not the game idea) and here filled out questionnaire.

During this workshop, 5 out of 10 game ideas were developed further. For that, personas and game design ideas were developed on the basis of the questionnaires. A persona is a description of user characteristics and her/his aims (Cooper, 2003). According to Cooper, who first introduced the concept of "persona" in the HCI community, a persona should be presented in text and/or image and it is usually generated to help designers to understand, describe, and define user preferences and behavior patterns. At the end of this workshop, their findings were presented per group, e.g., as a mock-up or a Pocket Code program. All groups had at least something to present at the end, either a concept and assets (2), persona (3), or half-finished Pocket Code programs (3). One group created their whole game with the programming engine "Processing"[93]. In the following section, the output per group will be presented. In addition, the author adapted or added missing personas and screenshots of the games.

Project 1: CatWalk

Game cards from focus group participant 10 (see Figure 5.15)

Group of 3 female university students[94]

[93]Processing: https://processing.org

[94]Moschik Maria, Mathis Eva, Schlacher Sabine

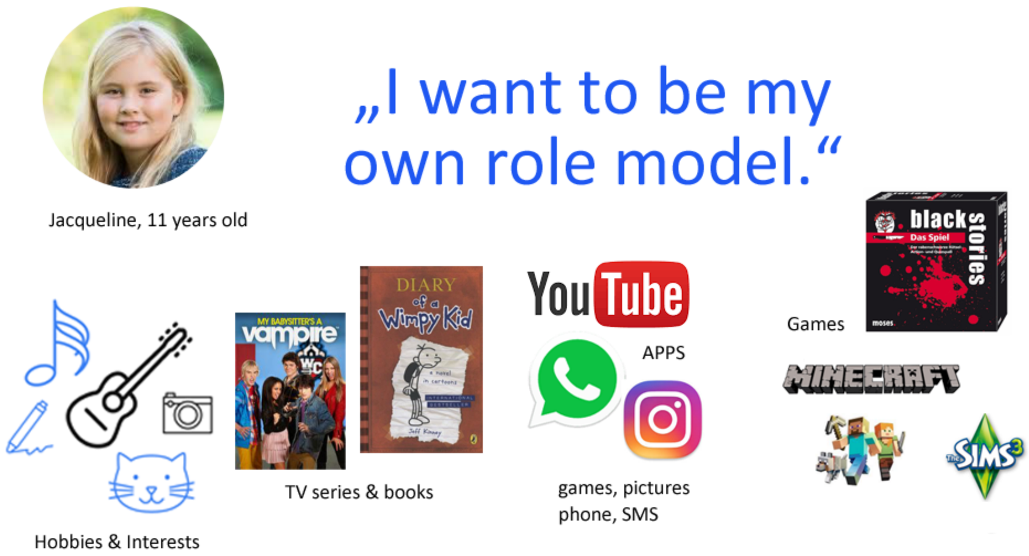


Figure 5.15.: Persona created for participant number 10.

Genre: Action; **Theme:** Horror; **Main character:** cat

Side character: mummies, crocodiles, cactus, godfather, tools; **Goal:** Catch, Solve

MD: levels/status, challenges/achievements; **A:** sensation, discovery

Story: A cat lands in the time machine of its owner, she starts in ancient Egypt and tries to get through different time periods back to her owner. She encounters enemies/obstacles from the time periods.

Similar games: Canabalt, Temple Run, Super Mario Run, Causality

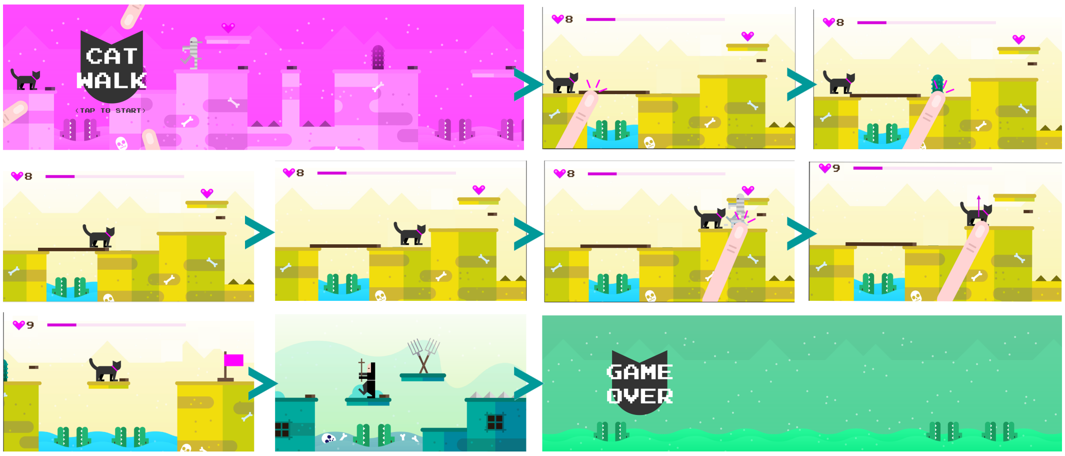


Figure 5.16.: Gaming concept for CatWalk.

This group provided the assets, the persona (adjusted by the author), the concept, and 2 half-finished programs in Pocket Code. The Pocket Code programs did not work correctly (the movement of the cat was not right).

Project 2: MagicAndMore

Game idea of focus group participant 8 (see Figure 5.17)

Group of 2 male and one female university students[95]

Figure 5.17.: Persona created for participant number 8.

Genre: adventure; **Theme:** fantasy; **Main character:** woman (ferry)

Goal: Territorial; **MD:** levels/status, challenges/achievements A: sensation, fantasy, narrative, discovery

Story: You can choose between three female characters at the beginning. You have to solve minigames at three islands. In the first level, you have to save the city from burning down. You need to conquer each island through minigames.

[95]Lackner Florian, Ortner Adriana, Wintschnig Michael

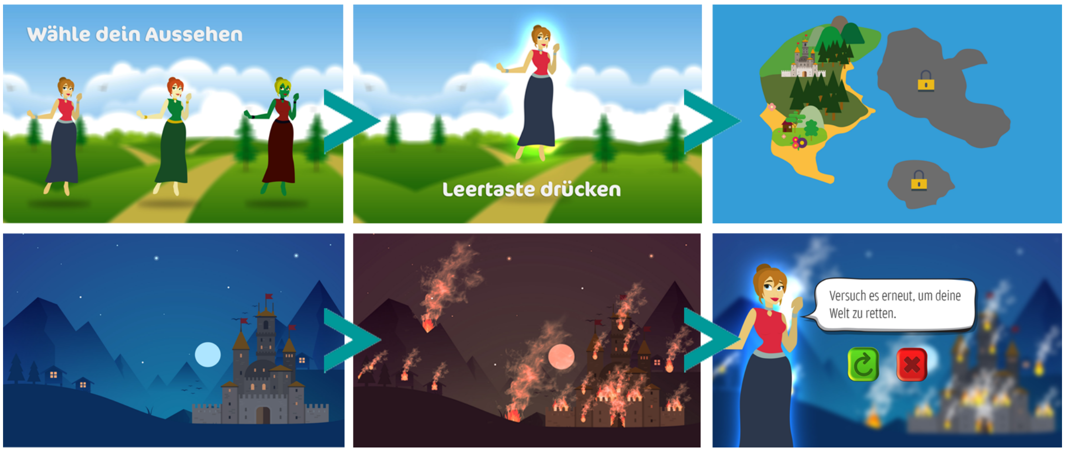


Figure 5.18.: Gaming concept for MagicAndMore.

Students who chose this gaming idea used the coding engine "Processing" and developed a fully functional game with it. The also provided the assets but no overall concept for the game and no persona for participant number 8.

Project 3: MelodicRider

Game idea of focus group participant 2 (see Figure 5.19)

Group of 2 female and one male university students[96]

[96]Kremser Alexander, Aron Larissa, Unteregger Sara

Figure 5.19.: Persona created for participant number 2.

Genre: jump&run; **Theme:** music, romance; **Main character:** skater girl

Goal: Capture; **MD:** points/reward, levels/status, challenges/achievements; **A:** challenge

Story: The world is dominated by music and skater girl has never left the village. She has "a special beat" and is looking for someone who has a similar one. She skates through unknown areas to get a feeling for rhythm.

Gaming world: colorful, cool, bright and friendly colors, music according to age, direction of electric guitar and electronic music, graffiti, roads, city, obstacles

Game logic: side scroller, flat design, every level has new music, points for right tones, new tricks, colors with each level, the more clashes the more points, jump = tap

Similar games: Beat Buddy, RayMan, Guitar Hero/Rock Band, Beat Sneak Bandits, Tony Hawk, Lego Island, Skate 3

Figure 5.20.: Gaming concept for Melodic Rider.

These group presented a concept with the gaming idea, persona (enhanced by the author), and all assets and screens. This group did not develop a game in Pocket Code.

Project 4: Princess Universe

Game idea of focus group participant 9 (see Figure 5.21)

Group of 3 female university students[97]

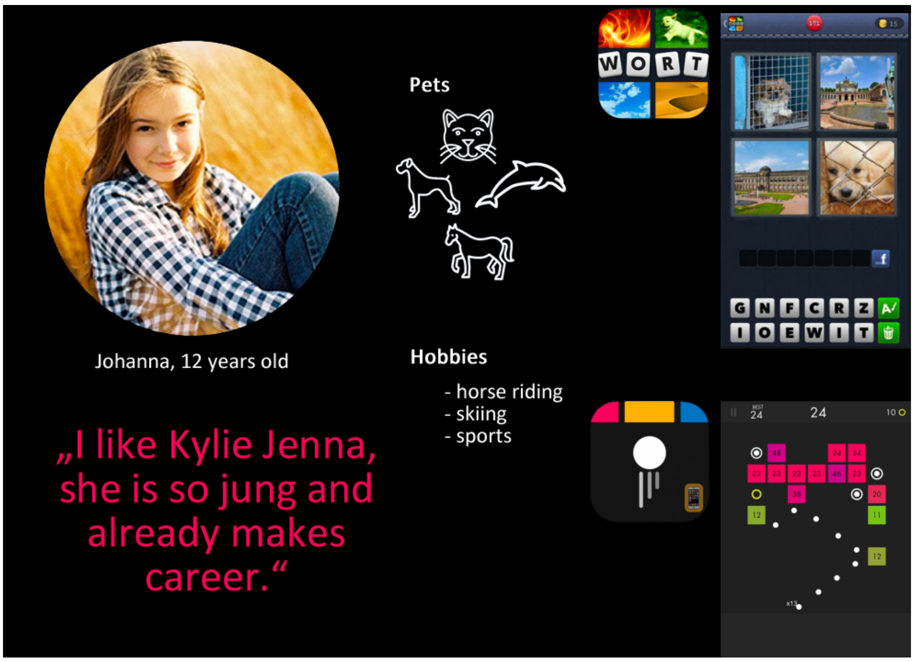


Figure 5.21.: Persona created for participant number 9.

[97] Moder Elisabeth, Rentschler Verena, Heckenast Nadine

Genre: action, quiz; **Theme:** space; **Main character:** princess-ferry

Goal: Capture/Avoid, Solve; **MD:** points/reward, levels/status; **A:** fantasy

Story: The main goal of her mission is to explore the universe and answer questions. The princess ferry visits different planets (Uranus, Saturn) and has to catch stars and lives and avoid the planets.

Game logic: button-up scroller; **Similar games:** Doodle Jump

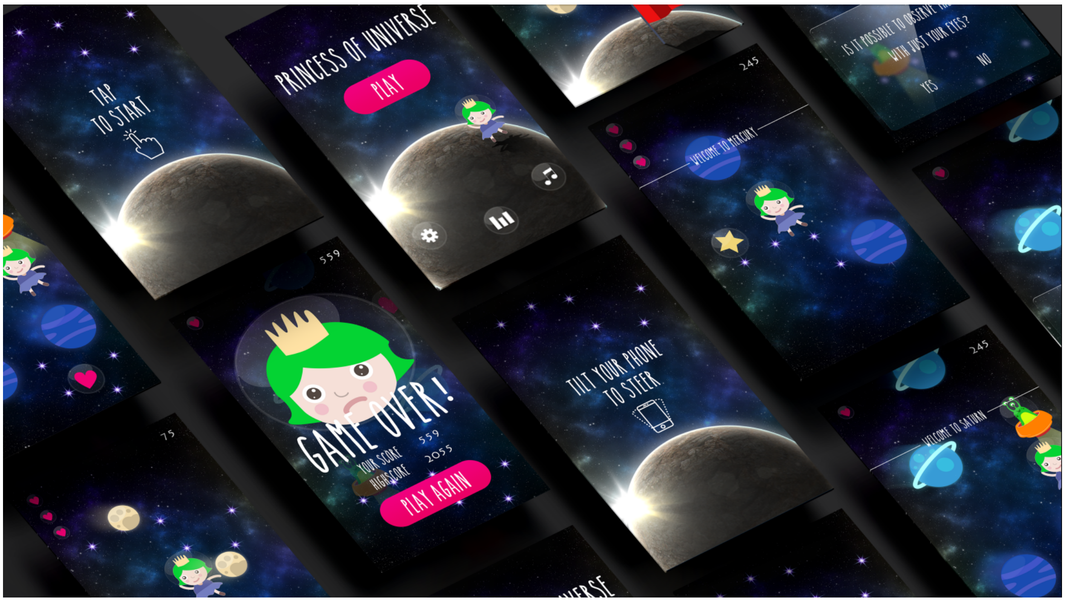


Figure 5.22.: Game concept of Princess of the Universe.

This group presented a persona (enhanced by the author), a short concept and all game assets. The group developed their game in Pocket Code but some functionalities were missing, e.g., no catching of stars or hearts, no questions and no end.

Project 5: Wendy & Randy

Game idea of focus group participant 5 (see Figure 5.23)

Group of 2 male university students[98]

[98]Sahin Christian, Wünscher Simon

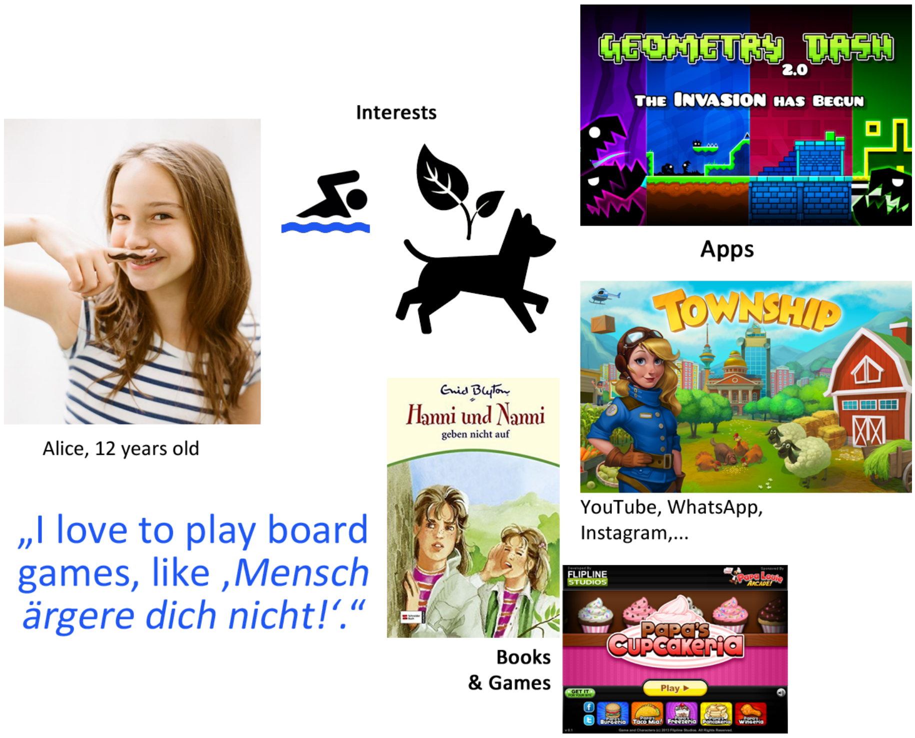


Figure 5.23.: Persona created for participant number 5.

Genre: jump&run; **Theme:** nature; **Main character:** dog, pony (unicorn)

Side characters: sunflowers; **Goal:** Collect; **MD:** points/reward; **A:** sensation, challenge

Story: Randy, a dog, has to catch sunflowers. Wendy the unicorn helps him at night and collects stars. Attention: There are small sandstorms. If you hit them Wendy disappears (and it is day again) and you lose a mushroom (life). The more sunflowers you catch the uglier the world around you becomes.

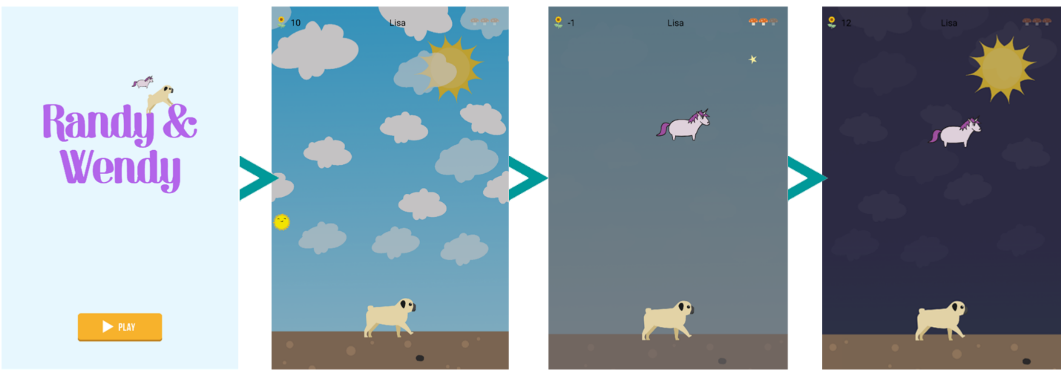


Figure 5.24.: Game concept of Wendy&Randy.

This group only submitted the game in Pocket Code and used the graphics of our media library. There is no end scene of the game (it ends if you reach 30 points) and the losing of lives has no consequence.

On the basis of concepts, assets, and half-finished Pocket Code games, the author started to develop the games in Pocket Code and to design banners for the community webshare view (with the help of Catrobat's design team).

5.3.2. Luna&Cat: a first concept to reinforce female teenagers.

Based on the focus group discussion, the PECC model helped as well to develop a concept for the girl's version of Pocket Code. Finally, the name chosen for this version is Luna&Cat. Although more of the girls voted for Make(IT)up, the name has an overall bad meaning. "To make something up" means to compensate for something or to repair something. Luna&Cat sounds nice and seems to be very appealing to our focus group. Within the logo, you see only a moon and a cat but obviously, Luna is the name of a girl, and her companion is the cat. One see Luna (the girl) in the Google Play banner. We asked the female students only to define a color for the community webshare. The reason is that in a planned future development users are allowed to choose between different themes in all apps, thus allowing customization. This should subsequently also affect the community webshare which already happened during the Alice Game Jam event. The violet used for that jam was very appealing and was rated very positively by the focus group as well. The color seems not to be too girly and is very friendly and bright. However, since the app has been described as too dark and blue, the first version of the app will be violet as well (without the option to switch colors). Figure 5.25 shows the new launcher icon, the themed UI of the app, and the new community webshare view with featured games. The second screenshot shows the new example game of Luna&Cat "penguin dance".

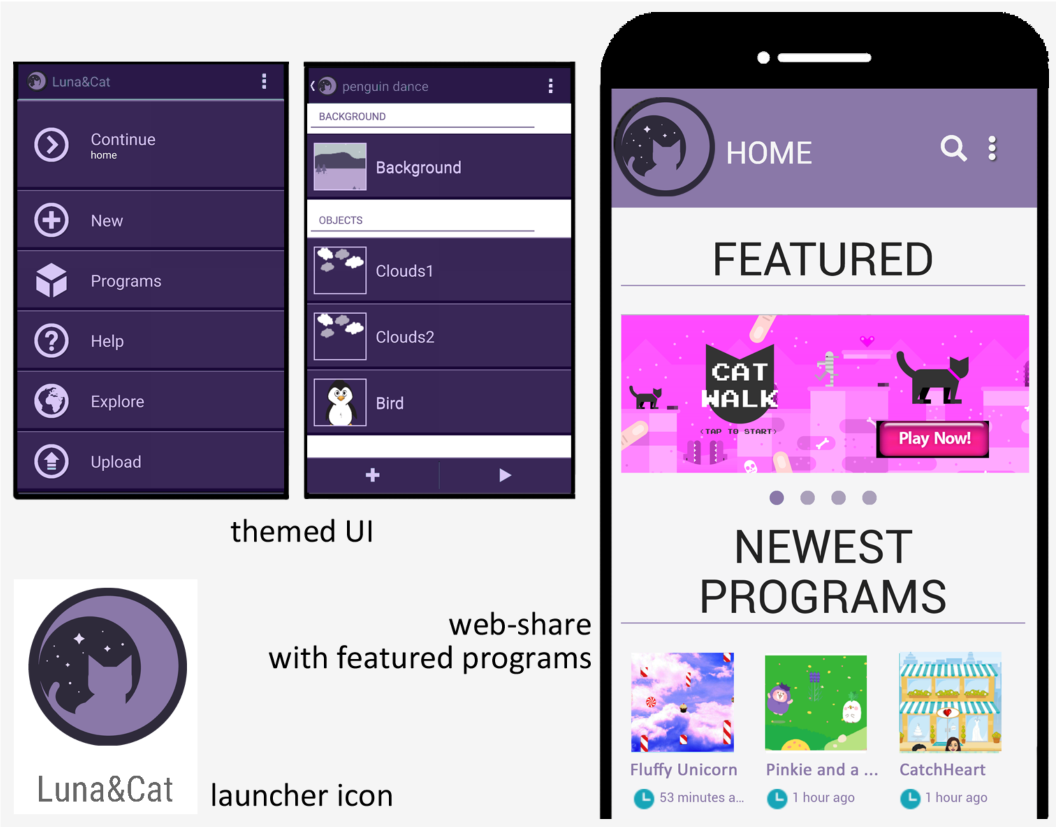


Figure 5.25.: Luna&Cat Logo, UI, and community webshare.

Luna&Cat is available at Google Play as a closed apk (see Figure 5.26). Thus, it is accessible for everyone who gets the permission to download it. In that way, the team could use the app for first usability tests and in August 2018 for a girls-only workshop (see Section 6.2). Since all school tablets have the same google account we can easily download the Luna&Cat beta apk on all tablets.

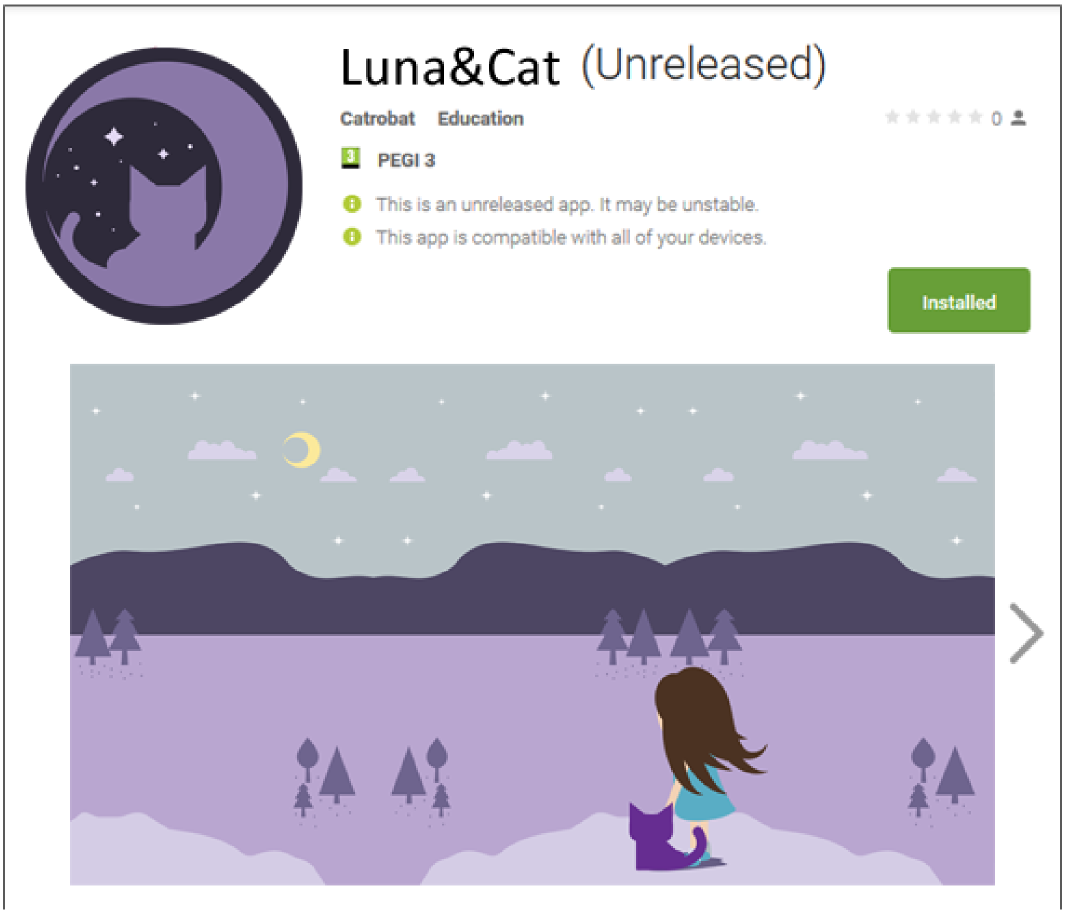


Figure 5.26.: Google Play Store: closed beta apk.

To optimize the store listing on Google Play, Google proposes to run experiments, for instance, to find the most effective graphics for the app (Google, 2018; Google Developer, 2018). Thus, developers are able to include variants of the app's icon, feature graphic, screenshots and promo videos. The app icon plays an important role in whether people choose the app from search results and category lists or not. Furthermore, the screenshots and brief descriptions for the icons are important, as users will see them when they select the app from the provided results list.

5.3.2.1. PECC model: to foster playing and engagement with Luna&Cat.

The literature review (see Section 2.4.2.4) and observations during NOLB showed the importance of female teenagers seeing games they like and letting them play. Designing gender-sensitive games is a promising approach to close the gender gap (Martinson, 2005). One study (Heeter et al., 2000) examined the topic and found that games designed by girls were ranked higher by female teenagers than by male teenagers. In addition, they considered them to be significantly better for learning than the boys' games.

During the Pocket Code units, there were several times that male students shouted through the class "Cool, there is a Star Wars/Alien shooter game" etc., whereas their female colleagues usually only played games created by their female peers or they played the example bird game. Currently, the recommended games in the community webshare sections are very similar, except for the random and new programs category. For newly uploaded programs, it is very difficult to reach the most downloaded/most viewed section at all. Currently, there are more than 45,800 programs and 1,252,460 program downloads[99]. Thus, we see that our users love to download and play existing games, create and design similar games, or remix existing ones (i.e., a game created on the basis of another game). For our users, it is a challenge to redesign popular games, and the community webshare is full of new versions of programs, like cookie catcher, ping pong, pacman, or skill games, like tic tac toe. To conclude, sharing is critical in the PECC model as well (see PECC Guidelines in Appendix A.4). Many successful apps have a very active community; see Pinterest, YouTube, or Instagram. The community is essential to feeling engaged. People who are part of this community want to feel a sense of belonging as well.

With Luna&Cat we can present the female teenagers more than just a nice app. Above all, we can give them appealing featured games created on basis of the ideas from our focus group. Four of the five games from the design workshop (see Section 5.3.1.3) have been adopted or refactored and are now available on the Luna&Cat community webshare of the app (see Figure 5.27). The game “Wendy&Rendy” has not been refactored, because the gameplay was not logical and the university students used graphics from our Catrobat Media Library. Some games such as “CatWalk” provide a scene with additional graphics to add a second level, or Magic&More an opportunity to integrate two more level. In addition, two more games have been chosen as featured games, either created by female participants of this year’s Google Code-In completion (“The boy who cried wolf”- a story telling game made by a female teenager) or from a girls group during NOLB (“Wer is der Mörder?”). Participant number 4 of the focus group discussion also mentioned that she want to have criminal/detective story game, thus we added this game to the Luna&Cat webshare view as well. The new featured games are fascinating, provide appealing assets, themes and interesting genres for female teenagers.

Figure 5.27.: Banner for the new featured games of Luna&Cat.

Moreover, a new starter game has been chosen for the app (see Figure 5.25). The starter game from Pocket Code shows a bird which is flying in the sky. This bird game has been adapted and replaced by the penguin from the Pocket Code family (the penguin is the most often chosen animal from the Pocket Code family by the target group) who is dancing instead of flying. Furthermore, the sky has been replaced by a violet hill background, which better suits the app as a whole. One important part of the PECC model states “Let students play”. With this small but meaningful change, girls should be encouraged to play and have fun.

[99] State of 02.02.2018

Another important action was a closer look at our available YouTube videos. Within the questionnaire, the focus group was asked how to they find or search for new apps. The preferred answer from all was "Through the App Store" (answers App Store: 9, friends:7, youtube:1, advertisements:1). This matches with the research: YouTube is the most popular form of media for students between 13-24 year-olds; 96% of that age group spends an average of 11 hours a week on watching online videos (DEFY Media, 2017). Men in general spend 44% more time on YouTube per month than women (Blattberg, 2015) and women have very stereotyped preferences (female-dominated YouTube categories: makeup & cosmetics, skin & nail care, or pop music, in contrast with men, who search more for games, gaming consoles, and sports). Videos on YouTube have the potential for presenting tutorials, step-to-step guidelines, games, and promotion videos. A lot of Pocket Code videos already exist[100], mostly created by Pocket Code users themselves (e.g., it was also a task during GCI to provide a video for the games that were created). Some videos are from teachers and facilitators, and some are from very young users from foreign countries like Korea, Japan or Russia. Recently, YouTube states the gender/age per viewed video. This showed us that our promotion video "Pocket Code-create your own games, directly on your phone!"[101] is mostly viewed by male users between 13 to 35 years old which exceeds the number of female users(see Figure 5.28).

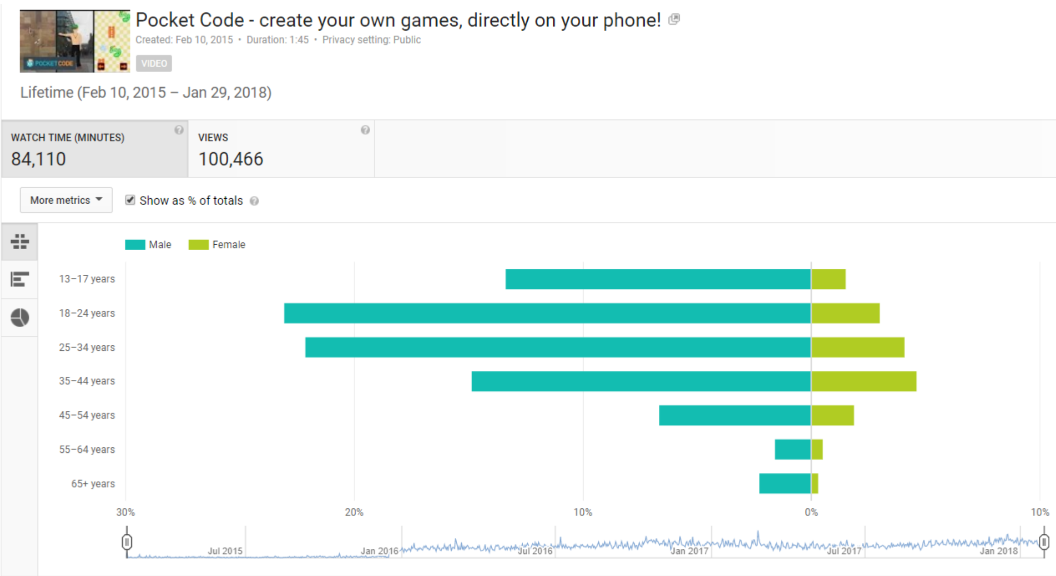


Figure 5.28.: YouTube statistic to "Pocket Code-create your own games, directly on your phone!" views per user/gender.

Considering this figure, it was very important for the author to create a new promotion video for our female target group (see Figure 5.29). This video is available at Google Play to give our female users first impressions about the app.

[100] Youtube Pocket Code search: https://www.youtube.com/results?search$_{q}uery = pocket + code$

[101] YouTube Pocket Code: https://www.youtube.com/watch?v=BHe2r2WU-T8

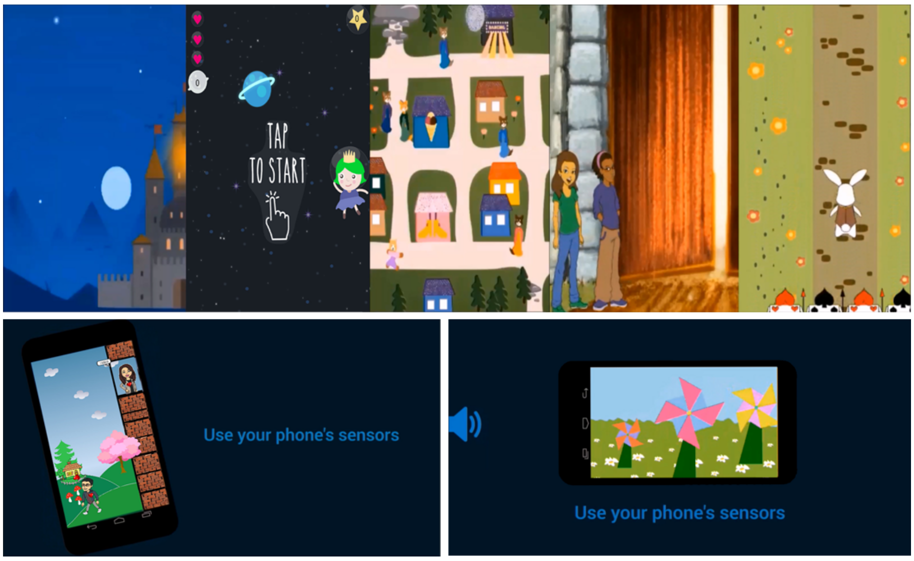


Figure 5.29.: New promotion video for Luna&Cat.

5.3.2.2. PECC model: to support creativity and coding with Luna&Cat.

To support female students' creativity, it is not only important to allow them to draw assets or to give them enough time during game design, but also to provide assets they can use, adopt, or even customize. This allows personalization, self-expression, and builds a sense of ownership. In addition, with professionally designed assets the games become much nicer, more beautiful, and more appealing. The NOLB courses and the PECC activity again show how important it is for female teenagers to have nice and appealing games of their own.

Thus, it is essential to expand our media library continuously. Assets like the one used in the RPG template allow the user to build own game characters, (see Section 4.3.5.2, Figure 5.30). Furthermore, assets from the new featured games are available for download in the media library and, in addition, Luna and the cat are available as well (see Figure 5.31).

Figure 5.30.: Catrobat Media Library: Build your own character.

By using the core functionality of the app — the coding — students can create games and animations with the use of a visual block based language. The authors research showed that no differences could be observed according the functional structure of students' games among genders (see Section 4.4.2). Comments concerning the app itself showed that users of all genders wish that the app would become more clearly structured and easier to use. Therefore, the whole Catroid team works at the time of writing this thesis on refactoring and redesign tasks to make the app more appealing for our whole target group.

To conclude, Luna&Cat has the same programming functionality as the default Pocket Code app. To support our female students in their coding, two missing functionalities that were mentioned in surveys, interviews, and focus group discussions by females in particular during NOLB and in Section 5.2 need our urgent attention:

- Provide whole media packages and predefined objects
- Provide functions and starter programs with tutorials

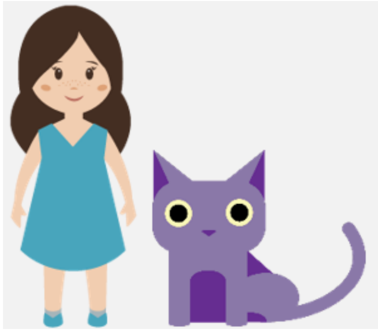


Figure 5.31.: For the Catrobat Media Library: Luna and her cat.

With the download of whole media packages, users can start with a new program which already integrates all assets, e.g., for creating a space themed game (backgrounds, objects with different looks, etc.). Thus, a perfect starting point is provided, to create personalized and sophisticated worlds, which is important to fulfil the creativity component of PECC. For many objects, several different looks exist within the media library to build movements or other kinds of animations. Currently, it is very time-consuming to download each look separately. Predefined objects can help to provide all looks for one object with the bricks, e.g., for the animation. The download of media packages is not only helpful for Luna&Cat but for all our apps. Within the Create@School app, we already provided templates that were especially designed for girls (see Section 3.4.2.2). The evaluation showed that these either were useful only for advanced users or cannot be used without guidance, i.e., outside school. Overall, the concept of the templates is good, but the templates by themselves make no sense for the default user outside school, like already discussed in the summary of NOLB (see Section 4.7). Each template is connected with a tutorial for teachers and students. For the default user who starts without help, easier starter programs must be provided. Currently, they are part of our tutorial page but there are some considerations to make them as part of the app, like the pre-coded templates in Create@School. The advantage is that if a user finds them easier, they can download them in the language of their phone, choose the preferred orientation, or merge them directly in one of their existing programs. If it is necessary to have a tutorial as well, a solution via web-view would be the most efficient. Thus, it is possible to swipe it in if you need it and out again. This concepts are currently in discussion and will be integrated after the refactoring work of our Catroid team.

In Scratch, users have the possibility to create their own customized bricks which act as subscripts (sub-functions). These kinds of blocks can have input like a string or a number. Userbricks in Pocket Code are currently in development but on-hold. Userbricks can provide predefined functions to merge into private projects or to share with others. Figure 5.32 shows a prototype in Pocket Code. Such blocks help with difficult coding parts to create more advanced programs without any guidance.

In addition, new websites developed during CS4HS 2017[102] are available online. The first website is an online course for tinkering with the Pocket Code[103] (focused on the use of mobile sensors), while the second one is a new educational website[104] with new brick documentations (with videos, examples, etc.), How-To's (for animations, effects, games, stories, and sounds), and other useful content. Therefore, a lot of the existing educational material was adopted and adjusted to be more similar to the useful Scratch tutorials. Each of these tutorials comes with a short starter program to download (see Figure 5.33).

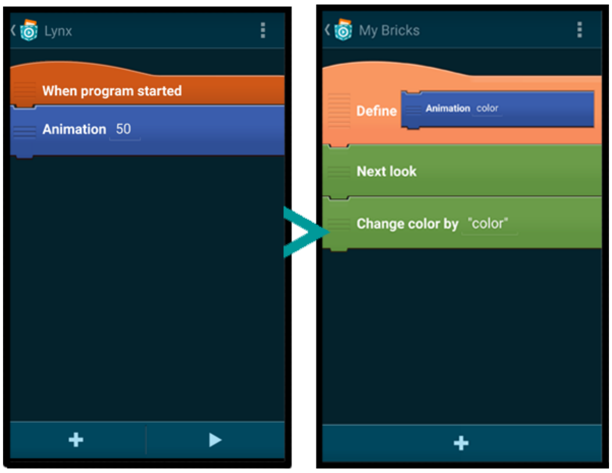


Figure 5.32.: "My bricks" in Pocket Code.

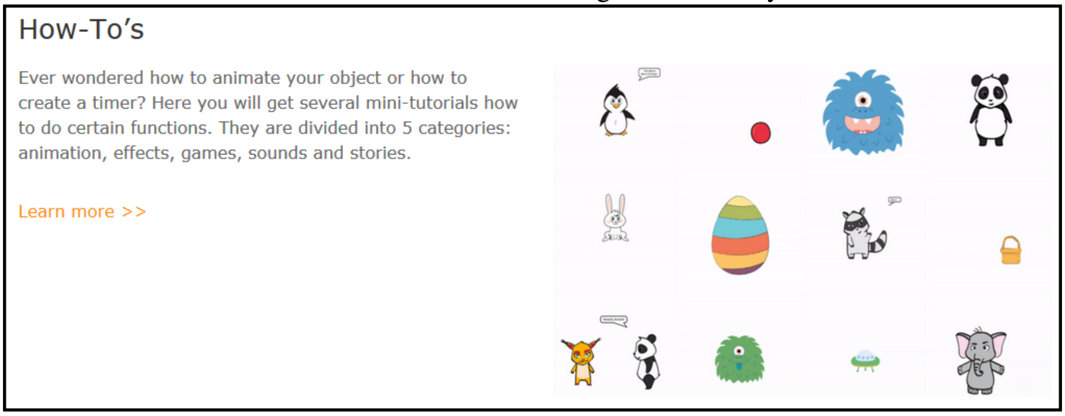


Figure 5.33.: New education website with useful "How to's".

To conclude, Luna&Cat should help, first, to build a new and large community of active female users, second, to provide interesting programs and an appealing environment, and third, to code individualized games by providing girls with ideas, tutorials, functionality, and more appealing assets.

5.4. Discussion of Results

The PECC model and guidelines have been developed and evaluated over a five month test period. Thus, the PECC has been applied to two courses to test if it is applicable for use to reinforce female teenagers through girls-only PECC-activities. With the help of a focus discussion group and design students, the new app Luna&Cat has been developed to be especially attractive for our female target group.

[102]CS4HS: https://cs4hs.cs.washington.edu/
[103]Catrobatblog: https://catrobatblog.wordpress.com/education/tinkering-with-your-phone/
[104]Catrobatblog 2: https://catrobatblog.wordpress.com/education/

5.4.1. Discussion RQ1.

The aim of Research Question 1 was to build the PECC model to organize playful, engaging, and creative coding activities to encourage (female) teenagers in computer science. The PECC model has been constructed to support different learning situations in coding, considers different approaches, and most of all it is gender-sensitive and not limited to a specific coding environment/tool. Thus, to apply it to different learning contexts and validating its applicability and usefulness, is highly desirable. PECC is based on relevant literature and findings about female teenagers (see Section 2.3 and Section 2.4). The author argues that it can be used to foster all genders but it is most importantly used to focus distinctly on girls who are starting to learn about coding. However, a more gender sensitive approach is indeed to provide benefits for all and one should not draw the conclusion that computer science could only be taught in gender homogeneous courses. The model should serve to make all its users (students, teachers, and facilitators) more gender sensitive. The literature argues (see Section 2.4.1.1) that female teenagers in particular feel a “lack of fitting in” in technical areas. Male students seems to struggle less with their sense of belonging to technical fields because our environment confirms them as “more suitable” for tech. Hence, computer science education in or outside of school plays a very important role for female teenagers. Either they have their doubts confirmed and exclude computer science completely, or are supported in order to become confident, motivated, and excited to develop new ideas and to take an important role in the digital world of tomorrow.

The goal of PECC is not to train all students to become game designers. It simply presents a playful approach of how to use coding while at the same time producing something that makes fun and most of all makes sense. For female teenagers it is important that tasks are goal oriented. Making a game is less challenging or motivating for them (as perhaps it is for boys) than telling them, for instance, to define a genre, theme, and a goal for games. Interviews and focus group discussions show that female teenagers have a sense for games and like to play. In a school setting, the risk is so much higher that students do not feel intrinsically motivated to design and create the games. Observations showed that male students more likely see such lessons as a welcoming and playful alternative to regular lessons than female students do. However, this is only true for higher grades (students older than 14 years). Grade 9 female students are more task and (learning) goal oriented than female students from lower grades. These observations are in line with the findings of the literature (see Section 2.4.1.4) and our results (see Section 4.4.3). Thus, the author proposes to start with PECC activities in lower grades. Experiences during NOLB and related literature suggest that the best age for starting with Pocket Code is in the seventh to eight grade (12 to 14 year olds).

In all STEM fields, it is indispensable to show and teach practical examples. As described in Section 2.2.2.2, the connection between playing/creating a game and future benefits is not as obvious as, for instance, conducting an experiment in physics class. Therefore, it is essential to draw the line to real life applications for jobs, and to define meaningful learning goals in respect to gender preferences.

Concerning the appropriate team sizes for coding activities, the author had mixed experiences. The best results could be achieved if students worked individually but with a common goal. In these classes the atmosphere was more relaxed, communication was more natural (meaning there was room for communication), and there was a greater feeling of ownership for the program that was created. However, in most IT professions and university projects, people are encouraged to work in teams. The ideal of the programmer who works on his own on one separate task is not the situation in reality. In reality, people are mostly part of a multidisciplinary team, have more responsibilities, and thus are able to gain knowledge in different fields and practices by switching projects and roles. Thus, teamwork should also be reinforced in PECC activities, but in a manner that is more guided and focused.

Gender sensitive language is important in all contexts; for example, to get a different perception of special professional groups, gender sensitive language must be considered in job advertisements or for the general workplace environment in an organization (see Section 2.4.1.2). However, gender related assignments start in the early childhood and are carried forward into kindergarten and junior levels. As a result, girls in high school are convinced that STEM is something for boys and teenage girls are grouped below a “hegemonic masculinity” society, which is seen as a cultural norm that connects men more to power and economic achievements (Connell, 1995). Thus, it is essential to be gender sensitive especially in STEM fields, to not foster stereotypes, and to assist students in their learning and production. It is not the aim of this thesis, in the scope of the wide range of this topic, to deal with all the issues merely touched upon by the brief description above. Gender sensitive education and gender awareness in schools is a relatively young research topic which is far from integrated into practice, and hardly any teachers are trained in gender contexts. In Section 2.4.1.2, the first considerations and the offering of gender trainings for teachers is discussed, but this is all only at the beginning of the process. In the future, the author will prepare and conduct such courses for students who will become computer science teachers (staring with trainers of the CoMaed courses at TU Graz).

In interviews, female students mentioned that they love to draw and to be creative, but considered themselves as not good in it. In each class, there are often one or two students who are good at drawing. It is very disappointing to begin with an idea in mind and end up with a game that looks very different. Thus, the author focused on providing new assets on a regular basis which cover different topics (e.g., Magic, Easter, Carnival), or to ask for new assets ideas. For instance, for course number 11 the group had the choice to let their needed artwork be designed by the Catrobat design team. Scratch, for instance, has used the same media library for more than 10 years and fosters their users in sharing their artwork. During GCI’17[105] we asked our community to do the same. Many graphics have been created but they are either not that professional, or made by boys, which means many images of cars, tools, etc. could be collected. Here again, the focus lies on providing a variety and also diversity of different graphics; it would be more ideal to represent different racial groups, cultures, or preferences in the future. Currently, the Catrobat Media Library is quite unstructured and one has to scroll down a huge list of different assets (with a bad internet connection this could be very time-consuming). Thus, the proposed media packages in Section 5.3.2.2 are one good solution.

The NOLB GMTF was very helpful to developing PECC. On the one hand, the GMTF, which focuses on teachers, refers to the different stages of the course (preparing, objectives, and assessment) and provides a structure. On the other hand, the model is more informative and theoretical than PECC, focuses mainly on the game development part, and is not applicable outside of school or to other coding tools. The GMTF covers specific steps important for schools, e.g., it considers school visions and beliefs, and curriculum adaptation and mapping. Thus, it can serve as an additional supportive framework for teachers, especially for those who have no technical background. The target group of the non-technical teachers was very important during NOLB as well as the assessment and the learning goal. The target group of PECC are any facilitators who want to use PECC activities in a more gender sensitive and extended area. The assessment is also part of PECC but only optionally. During NOLB, the team observed that most of the time teachers did not see the need to grade the students’ programs and some were just interested in if their students fulfilled the learning goal. With the NOLB PMD (see Section 3.4.2.5) the NOLB team strived to help teachers. Teachers who participated during NOLB in Austria did not see the value of the PMD. For them it was more informative and they did not use it for grading or further actions (e.g., adjustments based on the behavioral measurements). The PMD and

[105] Google Code In: https://codein.withgoogle.com/

the analytics tool was very interesting from a researcher's point of view, but needs more adoption to be helpful for teachers. Some of the proposed evaluations have been adopted for the PECC assessment sheet (see Appendix A.6). GMTF and its services were focused a lot on the game creation and on the final product itself. PECC in contrast also considers the introduction stage as important, whereas this is not a part in the GMTF.

PECC draws upon the GDBL approach or other models like PLE/CPL or the GERD model. PLE (see Section 2.2.2.1) supports the concepts of playfulness, collaboration, and creativity as well while focusing on emotions as a key role in thinking and learning. Furthermore, it structures the course in three parts: orientation lesson, game idea creation, playing phase. In contrast to PECC, the PLE concept does not focus on the game creation process but more on the creation of the idea. Thus, in their case studies, students (up to 10 years old) create their ideas in teams but never realize them with tools. However, it still fosters their creative thinking skills. PECC, in contrast, focuses more on the coding and designing process in practice. GDFs concentrate more on advanced coding environments and existing game development kits. GDBL also focuses on goal-oriented lectures, providing starter lessons, and collaboration, but is very focused on school environments and includes scoring, performance, and school limitations as well. This model also takes into account two forms of creating games (learning by creating games/modifying games) by setting the course aim and implicitly integrating the exchange of feedback and guidance between student and teacher. Thus, it again follows "rules" of the school system in a way similar to GMTF and is thus very focused on task completion and teachers. The GERD model (see Section 2.4.2.5) has been developed by Uni Bremen[106] but to the best of the author's knowledge, it has not be translated into English until now and is only available in German. This model integrates a lot of tasks explicitly which are only mentioned in the PECC guidelines, e.g., language and values, and it is more focused on people who have not already dealt with gender and diversity in their research or software development. Thus, it connects gender studies and computer science issues and how gender and diversity can be included. In addition, a "reflection catalog" is offered. This model targets researchers and developers, not teachers or students, but it has helped in integrating the gender aspects to PECC.

For PECC it was essential to provide a picture for the whole coding course and take into account the intrinsic and extrinsic motivators as well. Thus, PECC should serve as an applicable practical model for coding facilitators and environments.

5.4.2. Discussion RQ2.

The aim of Research Question 2 was to design girls-only PECC activities and to evaluate its positive influence in girls' performance and extrinsic/intrinsic motivation in regard to coding. This case study had some limitations. First, the sample was very small, second, the activity was part of the regular school units, which always compete with the intrinsic and extrinsic motivators (it would be beneficial to test PECC outside the school setting as well), and third, the game idea was not part of their curriculum (planets). However, the course provides a good starting point for further analysis and case studies.

Observations showed that girls from both groups were again very precise in their work (e.g., cutting out, placing their objects, adopting small steps to make their program better etc.). For these units, the analysis of the audio records were important for the author to identify differences in the questions

[106] GERD model: http://www.informatik.uni-bremen.de/soteg/gerd/?action=modell

asked. However, students in both groups collaborated less, did not help each other, and were more likely to ask the facilitator than their peers which was intentional to collect asked questions. This stood in contrast to the boys, who were more likely to try it out on their own. The classroom atmosphere was seen very positively by all in the test group. It was not noisy, everyone was concentrated on their work but at the same time communicated a lot, and it was very pleasant, whereas in the second units, the classroom climate in the control group was very noisy and more stressful. Another important observation was that the self-created template ("bricks on paper" activity) was mostly seen as very positive, and students had less questions during the coding units. However, some students from both groups only added brick by brick to their Pocket Code program (with the help of their template) without testing their program. Thus, they were left wondering at the end why the program did not work correctly. For future activities, it is important to provide them with more sub-goals, e.g., to test every step, and to not code the objects separately but in a parallel sessions (e.g., when using broadcast messages).

Even if the sample size was too small to show significant results (and the predefined hypothesis can not be validated) in the quantitative evaluation, the results are still interesting. In addition, the questionnaires that were developed have the potential to serve as an effective tool to measure intrinsic motivators of the PECC model in the future, thus they will be used for further research studies as well. The evaluation of the intrinsic motivator "Interest" showed that the interest level was higher in the test group than in the control group. Answers to the intrinsic motivator "Self-efficiency" let conclude that students of the test group had a higher confidence in using the app. Furthermore, they had the feeling they learned something new, and were more proud of their daily achievements (e.g., also of their brick on paper template), compared to students in the control group. This lets us assume that the test group placed value on their small steps/achievements and appreciated the learning factor of the app more. Collected factors related to the parameter "Sense of Belonging" showed that students of both groups learned something about technical professions and coding and hence showed slight increases in the feeling of importance of coding and sense of belonging. However, such a short course cannot change, e.g., a strong images of stereotypes or long held preconceptions. However, it is a starting point. The fun level was high in both groups but the intention to use the app further, or to partake in similar activities, was low. During the interviews, students said that they are not "the kind of person who codes". Here, a study conclude that students' enjoyment has no relation with their intention to participate in similar activities again (Giannakos et al., 2014). In the future, it would perhaps be more interesting to ask if they are willing to try such PECC activities outside school instead of during the regular lessons. To conclude, the quantitative evaluation shows that the values of many predictors for intrinsic motivations are located over the average. There are only small differences between the groups which shows the potential of the model to build successful learning environments for students of all genders. Students agreed, or strongly agreed that PECC fostered their interests, helped them learn something about coding, helped them gain better knowledge of technical professions and coding, and made them feel engaged while having fun during the course.

The answers from the open questions were interesting as well. For instance, the usability test during NOLB (see Section 4.2 and 4.3) showed that many of the male participants rated the aspects of coding and the learning effect of the app as positive whereas this was not a major point in the answers from participating girls. In contrast, in this study in the test group the coding aspect was mentioned almost completely by all girls. On the other hand, students from the control group wished to have more group work and instructions. For this course the engagement level and group work aspects needed more attention and should be fostered explicitly in further courses. During the interview, the participants in the test group stated that they were happy that they had the coding activity in a gender homogeneous class. The named reason was that the boys in their class were already experienced in coding and they might have laughed about their questions. This was a topic also in (Jaime and Ruby, 2011; Alvarado et al., 2017; Denner et al., 2015) and in Section 2.4.1. However, most of the coding classes in high school or university are mixed gender classes and even in girls-only classes, one will always encounter girls that are less or more motivated or insecure than others.

The program inspection and observations showed that students in both groups ran out of time. None of the students had time for the extra tasks, 78% achieved the academic learning goals (100% of the test groups) and 30% the learning goals regarding gaming. For integrating the end scene (as a part of the "Shape of a Game") variables were needed. Variables were not explained at the beginning but were to those who needed them. The team encouraged these students to explain it to the others. The problem was that the time for integrating the end scene was very short and not all achieved it (because they were still working on other parts). Perhaps one more double unit would be needed in order to provide the chance for all of the students to reach each of the learning goals as well as those in gaming. The program inspection again showed that students in the test group and girls from the control group were more concerned in generating nice programs (integrated more storytelling elements by alternate speech bubbles with emojis, etc.) and providing instructions. However, the evaluation of the quiz showed that students of the control group were able to answer more questions correctly compared to students from the test group, e.g., students from the control group could remember more categories and loops.

The case study showed the importance of the warm-up phase of PECC, which had not been considered during NOLB/in the GMTF. For the NOLB project, a more learning by doing approach without need of instructions and prior knowledge was used. During NOLB, some teachers had already expressed their concerns that they doubted that students understood coding or had learned something about coding. This correlates with the answers of interviews and observations, which showed that most of the students during NOLB did not understand the overall concept, they were left with open questions, and were happy that the course reached the end. The problem is that this warm-up task needs time, sensitivity, and preparation, but it is worth the effort. For future workshops, the author will foster more group discussions by working on these topics in small groups and discussing or presenting results. Coding is science and science is not something that can be learned without any theory. For example, nobody will teach about the anatomy of cats by letting students play with cats. Nevertheless, there are many unplugged coding and tinkering activities that can be used to present coding in a more fun and playful way. In the interview, a question about the bricks on paper activity was asked. Negative and positive comments could be collected. This shows that such activities can be helpful but it is not a concept for everybody (this is desirable but simple not possible). During the interview, girls from the test group had the idea to use the program just as an introduction for creating a bigger game, where they had more freedom of choice and personalization/customization opportunities. For instance, they suggested that their own avatars could visit the planets and have to solve different quests there. This is an interesting idea for future PECC activities and will be used for the project "RemoteMentor" (see next chapter, Section 6.2).

For the questionnaire, three more questions would be interesting to add. Since Li and Watson (Li and Watson, 2011) argue that by using block based/visual coding languages, students are often not aware that they are doing "real" coding, which is important for the self-efficacy motivational factor. Thus, it would be interesting to add the following three statements to future surveys:

1. By learning Pocket Code, I have been learning computer programming.
2. It will be easier for me to learn new programming languages because I know Pocket Code.

3. Catrobat is a computer programming language.

To conclude, the case study shows that the PECC model and the PECC guidelines have the potential to provide a checklist to build girls-only activities that:

1. motivate intrinsically, through fostering interest, self-efficacy, sense of belonging, and fun
2. motivate extrinsically, through being gender sensitive, allowing personalization/customization, defining appropriate (learning) goals, and games
3. help to reach the pre-defined (academic)/gaming learning goals

Thus, it allows to build playful, engaging, creativity and coding activities for girls-only initiatives. In the future, it has been planned to provide the guidelines at our educational platform to download for teachers and facilitators. Furthermore, in August 2018, a girls-only activity is planned together with CoMead to test PECC for activities outside school as well (see Section 6.2). This should help the author to refine, extend, and rework them together with teachers and facilitators. Further case studies will be performed with a higher number of participants to evaluate their significant potential and the defined four hypothesis in Section 5.2.3 to intrinsically and extrinsically inspire female students to engage in PECC activities.

5.4.3. Discussion RQ3.

Research Question 3 summarizes the customizations that are necessary in Pocket Code to provide teenage girls in PECC activities with an appropriate tool. Therefore, a new version of the Pocket Code app with the name Luna&Cat has been developed and uploaded as a closed beta apk to Google Play for first usability tests (i.e., store rating tests). For the development of these customizations, the PECC model was again very useful when defining in which areas Pocket Code could be customized in order to engage female students in coding activities. Therefore, the focus group discussion with teenagers from our potential user group was very useful and showed interesting insights. However, not all the results were considered in the development of the app. It is a fact that Catrobat, the name of our project, is not familiar to our users, and Scratch is also not used in all Austrian schools, thus the correlation to the name Luna&Cat was not obvious for the focus group. For them Make(IT)up brought it more to the point and showed a connection to IT/coding. Validate this app name (see Section 5.3.1.1) shows that the term has a more negative meaning in English. Furthermore, as described in Section 5.3.2.1 the app's presentation on Google Play is very important when attracting new users. Concerning Pocket Code, it is uncertain if female teenagers who search for games for girls will find or download it. We see a much more greater chance that the newly customized version Luna&Cat will be more appealing for girls and catch their attention, thus they will download it more often. Through optimization of the store listing on Google Play the chance is much higher to catch the preferred target group. The whole workflow for creating the featured games was very successful and beneficial: 1) inventing the game ideas together with the target group, 2) creating personas for the games, 3) designing the games during a workshop with "Industrial Design" university students, and, 4) finally developing the games. This workflow ensures that the story of the game is appealing to female teenagers, the visual design is sophisticated, and the final product fulfills functional requirements. The PECC model suggests improvements in all of its aspects and helped in developing Luna&Cat.

In Create@School, the team already implemented a different program view for female users. This concept was indeed to motivate the female users to become game developers on their own by showing them games made by girls and styles which they tend to like (see Section 2.4.2.4). As mentioned before, it is important for teenage girls to see that the community they are planning to join composed of people who have similar interests and predilections to feel a sense of belonging and self-efficacy. These integrated featured games were mostly subject-related and with Create@School we currently reach only a small user group (Create@School is still a beta version and not open for public use but can be used at schools if they request accounts). With Luna&Cat we hope that we can reach and build a new large user base of interested female teenagers who want to learn how to code. Usability tests and store listening tests in the future will help in our work. Therefore, the featured games and a new example game should encourage female students and awaken their interests for Catrobat programs. Playing these games should help them to get ideas and a sense for good game design through games of different genres, themes, stories, MDA, and characters that are attractive to them. As a result, they should feel supported and inspired to code their own programs or to remix existing ones. In the future, the author hopes that more and more games from female users will occupy the different sections in the web-view, like “Most Downloaded” or “Most Views”. Thus, these games will also be recognized by existing (female) Pocket Code users. As a result, the program collection will become more diverse and appealing to different user groups. Furthermore, the PECC model will help to support students in their engagement with Luna&Cat building a new increasing community of female teenagers who will love to share their games and assets with others, thus feeling a sense of belonging to our app/community. The possibility of liking and commenting programs and possibilities of remixing and downloading the games will cultivate a new, engaged female community from all over the world. Regarding the creativity aspect of PECC, with new assets and a continuously growing media library, the author strives to support female teenagers in creating appealing games. Thus, they are proud of their programs and lucky to share them with the community. For the future, new ways are needed on how to integrate assets that are coming from our (female) users and to value their work appropriate. The component of Coding in PECC should foster female students self-efficacy and provide them a tool for efficiently creating games. With the help of the new brick documentations and “How to’s” and in future, with the user bricks (which are at the time of writing still in development) students get more support during coding. In January 2018, a new project started with the aim of remotely mentoring female students. Thus, it is planned to train mentors of all genders and to support female students in their coding remotely via screen sharing options and chat functionality. This will help our female community in their first steps during coding (see Section 6.2). The tailored app Luna&Cat, the adapted community sharing platform view, the video, and the new “How to’s” programs should help to strengthen our female community from all over the world.

In conclusion, this chapter presented the results of this thesis: the PECC model and its guidelines (the theoretical output of this thesis) and provided results of its evaluation through conducting two case studies. Finally, the practical output of the thesis was presented: the new app for female teenagers Luna&Cat. Thus, all research questions could be answered. These gender sensitive customized services and tools will help female teenagers to reshape the technical domain in the future, thus playing an important role in closing the gender gap in ICT.

Chapter 6

Conclusion

The aim of this research was to identify the affordance necessary to foster female teenagers' intrinsic motivations with inclusive environments in computer science by Playing, Engaging, Creativity, and Coding (PECC). This chapter concludes with a recapitulation of the most essential findings and provides an outlook of the planned future activities and related future projects. The chapters of the theoretical part convey important background knowledge, first about the status quo in the CS education worldwide and in Austria, then about critical learning theories such as Constructionism. They also provide an overview about games and learning, design elements, and concepts. The second part of the literature review summarizes numbers and facts about women in technology, and further points out reasons for the gender gap in IT which has a dramatic impact on the career choices of female students in schools. Finally, this chapter suggested improvements for gender sensitive educational environments in coding. The status of (mobile) games for female teenagers is discussed and best practice examples from coding initiatives worldwide are presented. In the practical part of this thesis, the Catrobat apps as well as the NOLB project and its evaluation have been presented. This sets the basis for the further research of the author. As a result, a model (PECC) for gender equality in coding courses and underlying guidelines have been developed and evaluated in classrooms, as well as a new version Luna&Cat to target and reinforce female teenagers especially in coding activities.

6.1. Summary and Conclusion

Many of the changes in teaching and learning that resulted from studying the empowerment of women have improved the situation for all students. Within the NOLB project, the team assumed that learners who become game designers and creators would significantly contribute to closing the divide and participation gap in digital culture. All obtained results, which were presented in Chapter 4, showed that tailored coding activities and tools would be a great benefit for the female students.

During the Feasibility Study (see Section 4.2), the target group was examined. The findings suggest that girls play games less often than boys, and the genres of jump'n'run, skill, or puzzle games seem to be the most interesting for them. Challenges observed during the technology acceptance and usability analysis led researchers to conclude that the best solutions are to rethink group constellations (groups > 4 were less successful), to cut down the coding units to a maximal 4-5 double units, and to let students have more time to design their game ideas and assets, or to use more frameworks and set clear learning goals. This was especially critical for female students in order to scaffold the game design process but at the same time allow the freedom to personalize and customize their games to build a sense of ownership. This could help to reduce the stress level (overall, also for teachers) which was apparently very high during these first workshops. Observable differences showed that male students were more concerned with the gameplay itself, meaning the game should be fun, fast, and challenging. Girls mostly started with a rough idea. Next, they began designing and drawing their gaming worlds with a lot of care. This resulted in less time to code and get the game to a playable state. At the same time, the feasibility study suggests the importance of conformance and recognition of work done. The data show that the more students were motivated at the beginning, the more they were satisfied after the final Pocket Code unit. Thus, it is important that teachers and facilitators are conscious of their role as a mentor and motivator through the whole designing process.

For the Second Cycle of the project (see Section 4.3), a new version of Pocket Code, Create@School and corresponding services, offered new opportunities to be permanently integrated in school curricula. This version comes with a range of new features like physical engine integration, a more accurate and easier collision detection, speech/think bubbles for storytelling, new loops, usability improvements, and also integrates new features for schools. These new features, compromising of pre-defined templates, should be applicable for different subjects, thereby referring to several genres of action: adventure, puzzle, or quiz. This integration of important game elements makes the gameplay experience entertaining by using Mechanics, Dynamics, and Aesthetics (MDA). By tracking analytics data (see Section 4.3.2) within the app and the Project Management Dashboard (PMD) for teachers, the coding and design process become visible and accessible by teachers. Although, both features have been only used in the last three months of the project, interesting observations indicate a different behavior towards coding in males and females. The data coming from the app show that for female students, the social behavioral and self-regulatory aspects were the most important. The most notable differences between the genders, regarding the behavioral analysis, are visible in the parameters: creativity (female students used more of their own assets, and more often Pocket Paint for drawing or editing their assets), confidence (i.e. used more creation events — add/edit object, look, sound, and bricks), effort/dedicated time (female students spent more time either playing/testing their games or searching for other games), and interest (girls spent more time in the app overall). Observations showed that male students focused more on task completion, whereas it was more important for females to design sufficient programs/games.

In contrast to the differences in coding behavior, evaluation of the attractiveness of Create@School (see Section 4.3.1) was similar between the genders. The following similarities were noticed: both genders considered the app to be inventive, creative, innovative, and novel. However, the app, in general, has not yet achieved the desired acceptance level. Female students mentioned more often that the app separates them from people rather than brings them closer and found Create@School to be more isolating than connective. Female participants stated that they see Create@School as challenging. Conversely, the boys rated the app as undemanding. In spite of this, students showed an overall satisfaction while working with the app. The results of the questionnaire (see Section 4.3.3), which contains open-ended questions, show clear differences in answers between genders and highlights the females' desire for a suitable environment, where they feel included and are able to communicate, collaborate, and express themselves. Thus, female students prefer the designing, creativity, and freedom of choice aspects of the Create@School units. They did not mention the programming aspect explicitly. Additionally, the negative impressions from the female students concentrate more around the organization of the units or on the level of complexity and less on the app itself. Alternatively, the boys stated that the app was confusing and buggy. The focus group discussion with female students (see Section 4.3.4) showed that the app was confusing for them at some points, but they were more willing to ask for help or focus on problem-solving strategies encouraged by the teacher. Whereas boys were often frustrated with the app. In addition, the discussion suggests many interesting topics and elements to be important for this target group. The use of precoded templates (see Section 4.3.5) in class showed mixed results. They have been only tested in small groups. Points for improvement were recommended to make them more straightforward. Templates are a great resource to help in designing a game, but they are mainly developed for use in schools. Thus, it is the role of the teacher to adopt the templates to his or her subject and provide students with the information they need in order to start coding, but not continue with step-by-step instruction. This means, the templates are not intended to be used outside the classroom or for novice programmers. In the future, the plan is to integrate more starter or step-by-step programs in this menu option with important functionalities which will be included directly in the program.

Finally, observation of the submitted programs (see Section 4.4.2) showed that female students designed the most games within the genre adventure, text adventure, and puzzle games. Games designed by the boys were based on the genres of adventure (perhaps because it is an easy genre to achieve in Pocket Code), action shooter, and puzzle games. Whereas games by girls used more aesthetic components, boys used more advanced features of the app (e.g., physical properties) or high scores and challenges. Both groups use of main characters was very stereotypical: female students used mostly animals or fantasy characters; boys used transportation and male characters (transport items, e.g., spaceships and UFOs. Girls did not use these at all). However, no differences could be observed in the amount of bricks or other variables used in the games. The statistical reflection of the programs (see Section 4.4.3), in regard to the learning goal achievements, shows that female students are more willing to reach the predefined learning goal in lower grades, if they work separately on their games (tablets) and with the use of the pre-coded templates/frameworks. The findings are mostly consistent with related studies and suggest the optimal learning environment. Girls should be allowed a certain amount of self-expression, collaboration, and creativity. Evaluating game jams as a research method was a productive way during the NOLB project. The results of the two performed game jams (see Section 4.5), indicated that such events are more accepted by our male community or by teenagers who have already gained some coding experiences. This may be due to their competitive nature which builds more on extrinsic motivators, through awards, and less on self-efficacy.

The findings show that coding activities in school settings are well received but have underlying restrictions and the level of engagement and motivation can vary. This does not mean that coding should be banished from the academic curricula, quite the contrary is true. Special attention should be paid in terms of intrinsic (and extrinsic) motivational factors. Teachers should coach and mentor their students and especially encourage female teenagers in their classes (i.e., be more gender sensitive and aware). During the whole project, students' performance was not closely measured (e.g., grade them or test them on how to transfer knowledge from visual coding to more advanced programming languages). However, this was not the intention of the teachers who normally provide students with a grade or homework in CS subjects. The constructionism theory emphasizes more the process of knowledge construction than the actual resulting product (Papert, 1985).

Based on the results from the NOLB project and related literature, the author started to develop the PECC model to promote Playing, Engagement, Creativity, and Coding, with a special focus on gender sensibility during all its phases to reinforce (female) students as a result. Therefore, applicable guidelines for teachers and further templates for a storyboard, and assessment as well as an example lesson plan is provided. In addition to the four main components of the PECC model, four additional intrinsic and extrinsic motivators, have been identified to be the most promising to encourage especially females in coding activities. Facilitators and teachers can provide extrinsic reinforcement by providing (example) games the target group tends to like playing, but also by showing them underlying game principles, elements, and structures that are needed to create and design “good” games that flow fluidly and offer challenges. Games as a single motivator are not enough. It is also important for all students to indicate a direction through (learning) goals and subgoals, which prevent them from feeling stress, frustration, and anxiety.

Frameworks are promising extrinsic motivators that allow personalization and customization as well as the creation of individual assets that give a sense of ownership and pride. Furthermore, inclusion environments foster students to ask questions, to discuss openly, thus allowing communication and collaboration with gender sensibility and awareness in the language and examples used. More critical are the four identified intrinsic motivators: interest, sense of belonging, self-efficacy, and fun. The literature states that intrinsic motivators are more important for female students than they are for male students (see Section 2.4). Suggestions from the literature are to promote female role models, provide job clarity, include personal experiences, present IT as an interdisciplinary and creative discipline, to allow collaboration and communication, and to encourage female teenagers explicitly for these activities. These findings correspond with the NOLB findings. A first evaluation of PECC in a case study shows that the course setting encouraged all students, but students in the homogeneity classroom felt especially interested, had a higher confidence in using the app (self-efficacy), felt that they had learned something meaningful about IT professions, showed slight increases in their perception of importance of this field, and finally, students had fun.

The new app Luna&Cat has been developed on the basis of these findings and evaluated in reference to the PECC model, in terms of Playing, Engagement, Creativity, and Coding. It is integrated for “Playing” featured games developed together with two focus groups, design university students, and a customized example game. For “Engagement” this version in Google Play is presented to attract especially female teenagers, e.g., with a gender conscious new video and a new community where girls who like to code and design can meet girls with similar interests. For “Creativity” it includes new media assets, themes, and topics to create sophisticated games. Finally, for “Coding” new concepts will be included in all Catrobat apps for downloading media packages, in-app tutorials, starter programs, as well as customized user functions. The new brick documentation and “How to’s” should provide help.

Although the promising results of this thesis show that the developed approaches and tools provide a major possibility for female teenagers and schools to level improvements, they have still to prove its benefit in practice. Both concepts, the model and the app will be tested in August 2018, during a “Girls Coding Week”. An overview about future actions and further projects for the author, are presented in the next section.

6.2. Future Work

For the future, the author will continue working with female teenagers to reinforce them with coding activities. Thus, three activities are planned next:

- Girls Coding Week

The first coding event for girls-only was conducted in August 2017 as a first trial. The topic for this event was "Magic" since this seemed interesting for the target group. Although the author started to promote the event two months in advance, the flyer was shared via social media and printed in an offline and online newspaper the response was minimal. Because of this, two weeks before the event the registration had been closed. However, four girls attended the event and participated at the first "Girls Coding Day" event. All four girls were highly interested in creating games within the topic of "Magic" but the impact was very low. The author did not anticipate one important target group, which needed to be considered when regarding teenagers between 12 to 15 year old: the parents. In schools, the teachers are responsible for preparing the coding courses, out of school the parents need to be involved as well. For instance, parents who are employed need to plan care time and vacation trips for summer holidays almost a half year in advance. Whole day activities are received better than just afternoon activities, giving them the opportunity to drop of their children in the morning and pick them up again after work. Also, whole week activities are easier to plan than one day events. Furthermore, marketing and promoting are important to reach female teenagers who are not that likely to attend such workshops but should also address the parents. This reinforces, intrigues, and motivates them with attractive themes on flyers and posters that promote fun and creative activities (see Figure 6.1).

Figure 6.1.: Flyers and posters from the past to attract female teenagers for coding activities

The girls-only initiative at the TU Graz, CoMaed (see Section 2.4.2.5) provides summer camps for female teenagers between 10 to 15 years for several years and these courses are overbooked every year. This year a "Girls Coding Week" with the Catrobat apps is part of the CoMaed courses [108]. During this week, the author has the opportunity to apply the PECC model outside of the regular school setting, as well as test the Luna&Cat app and its featured programs with the main goal of reinforcing the female teenagers with new and engaging ways of learning. In addition, again a one week coding girls with a mixed group will serve again as a control group. For the future, it is planned to establish a "Girls Coding Club" for female teenagers-only at TU Graz to perform long-term studies and see an effect in the future. This club will be performed outside of the regular school setting and will especially emphasize creativity, allowing girls to explore IT and technology in a fun way to spark

[108] CoMaed course with Catrobat: https://www.comaed.tugraz.at/index.php/wbPage/wbShow/coding

their initial and long-term interest. Girls involved in the "Coding Clubs" will represent those who are vulnerable learners in terms of risk of exclusion. Using inclusive technology in a game-making environment is a revolutionary approach that will support social inclusion and empowerment from childhood to make a real change in the life of girls.

In addition, in June 2018, the author will complete a "gender-intersectionality-diversity" training. This will allow her, as a gender expert, to perform workshops in her field, first at her own institute to sharpen her colleagues for that important topic, and second at workshops for the trainer of the CoMaed courses (who are students in teaching) with the help of the PECC model. This will support them to gain a better understanding of gender sensitive computer science classes and to be more gender aware in these courses.

- Remote mentoring for female students[109]

This project "Remote Mentor", funded by NetIdee , Austria's largest Internet promotion campaign started in February 2018 and will conclude its result at the end of the year (see Chapter 3. The project was awarded in November 2017 with the "Closing the gender gap" award. First results of the remote mentoring workshops at schools are expected in June 2018.

Remote mentoring in Pocket Code should provide female beginners with real-time mentoring from advanced users in our community. A matchmaking system, modules for mentors and a bidirectional screen-sharing option, will support this process. Mentee and mentor can find each other online and the mentee will provide access to the code/Pocket Code. Via voice or chat channel with established systems additional support is provided. The mentors are first students of the TU Graz and later experienced users of the Catrobat community, who help female users especially in their first coding steps. Furthermore, it will be examined whether the gender of the mentors has an influence on the satisfaction and progress of the girls (to see positive influence of female role models). Therefore, we have partnered with sociology scientists from the University of Graz focusing on gender aspects for the research part. This team will focus on group discussions and interviews with female students.

- Stitching feature: New opportunities for handicraft lessons

To make the app more interesting and attractive for our target group of young women between 12 and 15 years old, in another project we are planning to extend the app with the option to program embroidery machines. In this way, self-created patterns and designs can be stitched on t-shirts, pants or even bags. Thus, with Pocket Code, the embroidery machines will be programmable. Patterns and different forms can be created using Pocket Codes' visual programming language. As a result, teenagers have something they can be proud of, something they can wear, and that they can show to others. For this purpose, a new version of Pocket Code for pattern embroidery will be developed. On the one hand this feature is intended to show young women new ways of using technology, with a lot of fun in a sustainable way. On the other hand, young men can get inspired through this digital design process to take part in textile handicraft lessons. Furthermore, appropriate stitching/coding courses will be offered at schools. A special emphasis will be given to a gender-equitable conception that considers different requirements, needs, and interests of our target group regarding to handicraft lessons. These courses will be realized together with "bits4kids"[110]. In cooperation with the fashion shop "Apflbutzn" [111] will help us to picture the whole workflow. This team will take part in the last units and bring their

[109] NetIdee RemoteMentor: https://www.netidee.at/remotementor
[110] bits4kids: http://www.bits4kids.at/
[111] Apflbutzn: http://www.apflbutzn.at/

embroidery machine. Thus, the teenagers are able to see how their programmed patterns are directly embroidered on shirts and bags. In addition, "Apflbutzn" will create an economical and sustainable concept, which can be integrated into existing (web-)shops and processes. In future, the pattern-files can be sent via mail and the embroidered products can be picked up or shipped.

The project's outcomes will be 1) the development of the stitch extension in Pocket Code, 2) a gender-equitable framework for stitch/coding courses, and 3) insights into the practical implementation for shops. As a result, these reseults will show how young women could be motivated by programming embroidery machines and, this concepts should serve as a guideline for others (Fab-Labs, schools, workshops, etc.).

The outcomes of this thesis, the PECC model and the Luna&Cat app, build the foundation for further improvement and projects to reinforce female teenagers for coding in the future. Further research projects, cooperation, and developments are aimed with the goal to close the gender gap at different levels of education.

Bibliography

ACKERMANN, E. 2001. Piaget's Constructivism, Papert's Constructionism: What's the difference? In *In Constructivism: Uses And Perspectives In Education*. 85–94.

ADAMS, M. 2018. Report: Users spent nearly $60 billion on apps in 2017. Retrieved February 08, 2018 from https://www.androidauthority.com/2017-app-revenue-827941/.

AGOSTO, D. 2003. Girls and gaming: A summary of the research with implications for practice. *In Teacher Librarian: The Journal for School Library Professionals 31,* 3, 910–931.

ALVARADO, C., CAO, Y., AND MINNES, M. 2017. Gender Differences in Students' Behaviors in CS Classes throughout the CS Major. *In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, 27–32. New York, USA.

APPIANING, J. AND ECK, R. N. V. 2015. Gender Differences in College Students' Perceptions of Technology-Related Jobs in Computer Science. *In International Journal of Gender, Science and Technology 7,* 1, 29–56.

APPOLICIOUS. 2017. Learning with Mobile Games. Retrieved February 08, 2018 from https://appolicious.com/learning-with-mobile-games/.

APPTORY. 2018. Know Your Target Audience: Mobile App Development Tips. Retrieved February 18, 2018 from https://fbombmedia.com/know-target-audience-mobile-app-development-tips/.

ARSETH, E. 2003. Playing Research: Methodological approaches to game analysis. *In Papers from spilforskning.dk Conference*, 1–8.

ASGARI, S., DASGUPTA, N., AND COTE, N. G. 2010. When Does Contact with Successful Ingroup Members Change Self-Stereotypes? *In Social Psychology 41,* 3, 203–211.

ASHCRAFT, C., MCLAIN, B., AND EGER, E. 2016. Women in Tech: The facts (2016 update). Retrieved March 20, 2018 from https://www.ncwit.org/sites/default/files/resources/womenintech_facts_fullreport_05132016.pdf. National Center for Women & Technology (NCWIT).

ATLANTA PARENT EDITORIAL. 2017. Pink, Blue & Neutral: How Gender Stereotypes Shape Kids. Retrieved March 20, 2018 from https://www.atlantaparent.com/pink_blue_neutral/.

AUSTRALIAN CURRICULUM, ASSESSMENT AND REPORTING AUTHORITY (ACARA). 2017. Learning area. Retrieved February 8, 2018 from http://www.australiancurriculum.edu.au/technologies/introduction.

BACK, J., CURZON, P., MYKETIAK, C., MYKETIAK, C., MCOWAN, P., AND MCOWAN, P. 2011. A study in engaging female students in computer science using role models. *In Proceedings of the 16th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, 63–67.

BACKLUND, P. AND HENDRIX, M. 2013. Educational Games-Are They Worth the Effort? a Literature Survey of the Effectiveness of Serious Games. *In 5th International Conference on Games and Virtual Worlds for Serious Applications*, 1–8.

BALANSKAT, A. AND ENGELHARDT, K. 2015. Computingour future. Computer programming and coding Priorities, school curricula and initiatives across Europe. *In European Schoolnet (EUN Partnership AIBSL)*.

BARBER, B., STONE, M., HUNT, J., AND ECCLES., J. S. 2005. Benefits of activity participation: The roles of identity affirmation and peer group norm sharing. *Organized activities as contexts of development: Extracurricular activities, after-school and community programs*, 185–210.

BARKER, L., HOVEY, C., AND THOMPSON, L. 2014. Results of a large-scale, mulit-instituional study of undertraduate retention in computing. *In Frontiers in Education Conference, 2014 IEEE*.

BARNETT, L. 2011. If it wasn't for Hedy Lamarr, we wouldn't have Wi-Fi. Retrieved March 20, 2018 from `https://www.theguardian.com/theguardian/shortcuts/2011/dec/04/hedy-lamarr-wifi`.

BARNETT, T., LAWLESS, B., KIM, H., AND VISTA, A. 2017. Complementary strategies for teaching collaboration and critical thinking skills. Retrieved February 08, 2018 from urlhttps://www.brookings.edu/blog/education-plus-development/2017/12/12/complementary-strategies-for-teaching-collaboration-and-critical-thinking-skills/. Education Plus Development.

BARTILLA, A. AND KÖPPE, C. 2016. Organizational Patterns for Increasing Gender Diversity in Computer Science Education. *In Proceedings of the 10th Travelling Conference on Pattern Languages of Programs* 10.

BAUER, I. 1995. Frauen, Männer, Beziehungen. Sozialgeschichte der Geschlechterverhältnisse der Zweiten Republik. *In J. Burger & E. Morawek (Hrsg.), Entwicklungslinien der Zweiten Republik 1945-1995*, 102–118.

BAUMEISTER, R. AND LEARY, M. 1995. The need to belong: desire for interpersonal attachments as a fundamental human motivation. *In Psychol Bulletin 17,* 3, 497–529.

BAUSTÄDTER, B. 2016. Wie (geschlechts)neutral kann Technik sein? Retrieved February 26, 2018 from `https://www.tugraz.at/fileadmin/user_upload/tugrazInternal/News_Stories/Print/people/pdf/160323_TUG_People_2016_03_web.pdf`. TU Graz People Magazin.

BEAM, M., CHEN, C., AND GREENBERGER, E. 2002. Thee nature of adolescents' relationships with their "very important" non-parental adults. *In American Journal of Community Psychology 30,* 2, 305–325.

BECKWITH, L. AND BURNETT, M. 2004. Gender: An important factor in end-user programming environments? *In IEEE Symp. Visual Lang. Human-Centric Comput.Lang. Environments*.

BECKWITH, L., BURNETT, M., GRIGOREANU, V., AND WIEDENBECK, S. 2006. Gender HCI: What about the software? *In IEEE Computer Society Press*.

BEEBOM. 2017. 10 Best Educational Apps for Android To Learn Almost Anything. Retrieved February 09, 2018 from https://beebom.com/best-educational-apps-android/.

BELLOTTI, F., BERTA, R., GLORI, A. D., OTT, M., AND ARNAB, S. 2011. Designing Serious Games for education: from Pedagogical principles to Game Mechanisms. In In Proceedings of the 5th European Conference on Games Based Learning. *In Proceedings 5th European Conference on Game-Based Learning*, 24–34.

BELTRÁN, M., URSA, Y., PETRI, A., SCHINDLER, C., SLANY, W., AND SPIELER, B. 2015. Inclusive gaming creation by design in formal learning environments: "girly-girls" user group in No One Left Behind. *In International Conference of Design, User Experience, and Usability. Design, User Experience, and Usability: Users and Interactions*, 153–161.

BELTRÁN, M. E., URSA, Y., SLANY, W., COLLAZOS, P., BROWN, D., AND BURTON, A. 2017. D6.7 — "No One Left Behind" Business Plan. *EU Delivery Agreement 645215*.

BENDER, W., ULIBARRI, D., AND KHANDELWAL, Y. 2016. Music Blocks: A Musical Microworld. *In Proceedings of Constructionism 2016*, 109–116. Bangkok, Thailand.

BERGSTRÖM, K., BJÖRK, S., AND LUNDGREN, S. 2010. Exploring aesthetical gameplay design patterns: camaraderie in four games. *In Proceedings of the 14th International Academic MindTrek Conference: Envisioning Future Media Environments*, 17–24.

BERKLEY GRADUATE DIVISION. 2018. Overview of Learning Theories. Retrieved February 08, 2018 from http://gsi.berkeley.edu/gsi-guide-contents/learning-theory-research/learning-overview/.

BERRI, J., BENLAMRI, R., AND ATIF, Y. 2006. Ontology-based framework for context-aware mobile learning. *In Proceedings of the 2006 international conference on Wireless communications and mobile computing*, 1307–1310.

BERS, M., FLANNERY, L., KAZAKOFF, E., AND SULLIVAN, A. 2014. Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *In Computers and Education 72*, 145–157.

BEVANS, A. 2017. Who plays mobile games? Retrieved February 08, 2018 from https://www.gamesindustry.biz/articles/2017-06-14-who-plays-mobile-games.

BEYER, S., CHAVEZ, M., AND RYNES, K. 2002. Gender differences in attitudes toward and confidence in computer science. *In Procedeengs of the Midwestern Psychological Association Annual Meeting, May 2002*.

BEYER, S., RYNES, K., PERRAUL, J., HAY, K., AND HALLER, S. 2003. Gender differences in computer science students. *In SIGCSE Bull 35,* 1, 49–53.

BHATTACHERJEE, A. AND PREMKUMAR, G. 2004. Understanding Changes in Belief and Attitude toward Information Technology Usage: A Theoretical Model and Longitudinal Test. *In MiS Quarterly 25,* 2, 229–254.

BINDER, D., THALER, B., UNGER, M., ECKER, B., MATHÄ, P., AND ZAUSSINGER, S. 2017. MINT an Öffentlichen Universitäten, Fachhochschulen sowie am Arbeitsmarkt - Eine Bestandsaufnahme. *Projektbericht Research Report, Studie im Auftrag des Bundesministeriums für Wissenschaft, Forschung und Wirtschaft (BMWFW).* http://irihs.ihs.ac.at/4284/1/2017-ihs-report-binder-mint-universitaeten-fachhochschulen.pdf.

BLATTBERG, E. 2015. The demographics of YouTube, in 5 charts. Retrieved February 25, 2018 from https://digiday.com/media/demographics-youtube-5-charts/.

BLIKSTEIN, P. AND KRANNICH, D. 2013. The Makers' Movement and FabLabs in Education: Experiences, Technologies, and Research. *In Proceedings of the 12th International Conference on Interaction Design and Children*, 613–616.

BOHYUN, K. 2015. Understanding Gamification. *Library Technology Reports 51,* 2, 17–20.

BOSCH, N. V., RAMOS, A. M. G., AND SAMARANCH, E. A. 2014. Doing and Undoing Genders and Information and Communication Technologies. *In Proceedings of the XV International Conference on Human Computer Interaction 80*.

BOTHA, A., HERSELMAN, M., AND VAN GREUNEN, D. 2010. Mobile user experience in a mlearning environment. *In Proceedings of the 2010 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists*, 29–38.

BOUCHER, L. 2016. What are the different types of population pyramids? Retrieved March 20, 2018 from https://populationeducation.org/content/what-are-different-types-population-pyramids. Population Education.

BOULTON, H., SPIELER, B., PETRI, A., SLANY, W., SCHINDLER, C., AND BELTRÁN, M. E. 2016. The role of game jams in developing informal learning of computational thinking: a cross-European case study. *In Proceedings of the 8th International Conference on Education and New Learning Technologies*.

BOULTON, H., TINNEY, J., BROWN, D., MARTINOVS, D., AND BELTRAN, M. E. 2016. D2.1 — Alignment of lesson plans and cognitive & behavioural developments. *EU Delivery Agreement 645215*.

BRACKMANN, C. P., ROMÁN-GONZÁLEZ, M., ROBLES, G., MORENO-LEÓN, J., CASALI, A., AND BARONE, D. 2017. Development of Computational Thinking Skills through Unplugged Activities in Primary School. *In Proceedings of the 12th Workshop on Primary and Secondary Computing Education, Erik Barendsen and Peter Hubwieser (Eds.)*, 65–72.

BRAND, S. AND DALTON, E. 2012. Universal Design for Learning: Cognitive Theory into Practice for Facilitating Comprehension in Early Literacy. *Forum on Public Policy Online 2012,* 1, 1–19.

BRATHWAITE, B. AND SCHREIBER, I. 2009. Challengesfor Game Designers. *Course Technology PTR: Stacy L. Hiquet*.

BRAUN, F. 2008. Mehr Frauen in der Sprache. Leitfaden zur geschlechteren Formulierung. *Ministerium für Justiz, Frauen, Jugend und Familie des Landes Schleswig Holstein*.

BROPHY, J. 2013. Motivating students to learn. *In New York, NY: Routledge*.

BUFIN, S., JOHNSON, N., AND GIBUM, C. 2015. Critical Perspectives on Technology and Education. *In. St. Martins Press LLC*.

BUNDESGESETZBLATT I NO. 66/2004. 2004. Bundesgesetz Ãijber die Gleichbehandlung (Gleichbehandlungsgesetz - GlBG). Retrieved March 20, 2018 from https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=20003395.

BURMESTER, M. AND KOLLER, F. 2008. Der User Experience (UX) auf der Spur: Zum Einsatz von www. attrakdiff. de. *In Usability Professionals*, 78–82.

BURN-CALLANDER, R. 2014. Just one in 20 job applicants for new roles in IT are women. Retrieved March 20, 2018 from `http://www.telegraph.co.uk/finance/jobs/11307481/Just-one-in-20-job-applicants-for-new-roles-in-IT-are-women.html`.

BURNETT, M., FLEMING, S., IQBAL, S., VENOLIA, G., RAJARAM, V., FAROOQ, U., GRIGOREANU, V., AND CZERWINSKI, M. 2010. Gender differences and programming environments: Across programming populations. *In ACM Empirical Software Engineering and Measurement*.

BUSKENS, I. AND WEBB, A. 2014. Women and ICT in Africa and the Middle East: Changing Selves, Changing Societies. *In Zed Books*.

CADINU, M., MAASS, A., ROSABIANCA, A., AND KIESNER, J. 2005. Why Do Women Underperform Under Stereotype Threat?: Evidence for the Role of Negative Thinking. *Psychological Science 16,* 7, 572–578.

CALLIOPE GMBH. 2017. Unsere Mission. Einen spielerischen Zugang zur digitalen Welt zu ermöglichen. Retrieved January 07, 2018 from `https://calliope.cc/mission`.

CAMPBELL, K. AND VERA, A. M. 2010. Female board appointments and firm valuation: Short and long-term effects. *In Journal of Management and Governance 14,* 1, 37–59.

CANNINGS, T. AND STAGER, G. S. 2003. Online Constructionism and the Future of Teacher Education. *In Proceedings of the 3.1 and 3.3 working groups conference on International federation for information processing: ICT and the teacher of the future 23*, 13–15.

CARTER, D. A., SIMKINS, B. J., AND SIMPSON, G. 2003. Corporate Governance, Board Diversity, and Firm Value. *In The Financial Review 38,* 1, 33–53.

CARTER, J. AND JENKINS, T. 1999. Gender and programming: what's going on? *In Proceedings of the 4th annual SIGCSE/SIGCUE ITiCSE conference on Innovation and technology in computer science education (ITiCSE '99)*, 1–4. New York, USA.

CARTER, L. 2006. Why students with an apparent aptitude for computer science don't choose to major in computer science. *In Proceedings of the 37th SIGCSE technical symposium on Computer science education*, 27–31.

CAST, INC. 2014. What is UDL?, Natural Center on Universal Design for Learning. Retrieved February 09, 2018 from `http://www.udlcenter.org/aboutudl/whatisudl`.

CASTILLO, R., GRAZZI, M., AND TACSIR, E. 2014. Women in Science and Technology. What Does the Literature Say? *Inter-American Development Bank. Institutions for Development* IDB-TN-637.

CATALYST. 2016. Women in Science, Technology, Engineering, and Mathematics, 2016. Retrieved September 10, 2017 from `http://www.catalyst.org/knowledge/`.

CHANCE, S. M. AND BOWE, B. 2015. Influence of Collaborative Learning on Women's Experiences of Engineering Education. *In Research in Engineering Education Society (REES)*.

CHANDRASEKARAN, S., STOJCEVSKI, A., LITTLEFAIR, G., AND JOORDENS, M. 2012. Learning through Projects in Engineering Education. In In Proceedings of the 40th SEFI Annual Conference. *SEFI 40th annual conference*, 126–134. ISBN= 9782873520052.

CHARLES, M. AND BRADLEY, K. 2009. Indulging our gendered selves? sex segregation by field of study in 44 countries. *In American Journal of Sociology 114*, 924–976.

CHARLESTON, L.-J. 2017. Ways The Tech Industry In Australia Is Trying To Attract And Retain Females. Retrieved March 04, 2018 from `https://www.huffingtonpost.com.au/2017/03/31/ways-the-tech-industry-in-australia-is-trying-to-attract-and-ret_a_22019419/`.

CHARSKY, D. 2010. From Edutainment to Serious Games: A Change in the Use of Game Characteristics. *In Games and Culture - Game Cult 5,* 2, 177–198.

CHASE, J. AND OKIE, E. 1999. Combining cooperative learning and peer instruction in introductory, computer science. *In SIGCSE Technical Symposium on Computer Science Education*.

CHATHAM, A., PIJNAPPEL, S., AND MUELLER, F. 2013. Game Jam. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*. 3175–3178. http://doi.acm.org/10.1145/2468356.2479640.

CHEN, J. 2007. Flow in games (and everything else). *In Communications of the ACM 50,* 4, 31–34.

CHERYAN, S. 2012. Understanding the paradox in math-related fields: Why do some gender gaps remain while others do not? *In Commentary. Sex Roles 66*, 184–190.

CHERYAN, S., MASTER, A., AND MELTZOFF, A. 2015. Cultural stereotypes as gatekeepers: Increasing girls' interest in computer science and engineering by diversifying stereotypes. *In Frontiers in Psychology 6,* 49.

CHERYAN, S., PLAUT, V., DAVIES, P., AND STEELE, C. 2009. Ambient belonging: How stereotypical cues impact gender participation in computer science. *In Journal of Personality and Social Psychology 97*, 1045–1060.

CHERYAN, S., PLAUT, V. C., HANDRON, C., AND HUDSO, L. 2013. The stereotypical computer scientist: Gendered media representations as a barrier to inclusion for women. *In Springer Science+ Business Media 69,* 1-2, 58.

CHERYAN, S., SIY, J. O., VICHAYAPAI, M., DRURY, B. J., AND KIM, S. 2011. Do female and male role models who embody STEM stereotypes hinder women's anticipated success in STEM? *In Journal of Personality and Social Psychology 2*, 656–664.

CLAPPER, T. 2009. Moving away from teaching and becoming a facilitator of learning. Professionals Against Improperly Labeling Active Learne. *PAILAL Newsletter 2*, 1–6.

CLARKE-MIDURA, J., ALLAN, V., AND CLOSE, K. 2016. Investigating the Role of Being a Mentor as a Way of Increasing Interest in CS. *In Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, 297–302.

CLAUDE, D., WAJDA, K., AND MAASS, S. 2014. GERD - Ein Vorgehensmodel zur Integration von Gender & Diversity in die Informatik. *In Vielfalt Informatik. Ein Betirag zu Selbstverständnis und Außenwirkung. Universität Bremen*.

CLAYTON, K. L., VON HELLENS, L. A., AND NIELSEN., S. H. 2009. Gender stereotypes prevail in ICT: a research review. In Proceedings of the special interest group on management information system's . *In Proceedings of the special interest group on management information system's 47th annual conference on Computer personnel research*, 153–158.

CO, P. 2006. Level design for games, creating compelling game experiences. *Berkeley, New Riders Games*.

CODE.ORG. 2018. Hour of Code. Retrieved January 18, 2018 from `https://hourofcode.com/de`.

COHOON, J. 2011. Promising Practices: Encouragement Works in Academic Settings (Case Study 1). *Boulder, CO: National Center for Women & Information Technology*.

COLLAZOS, P., PÉREZ, F., BELTRÁN, M. E., GAETA, E., CEA, G., AND SPIELER, B. 2017. D4.3 — Integration of Analytics and Dashboard. *EU Delivery Agreement 645215*.

COLLAZOS, P., REDONDO, T., BELTRÁN, M. E., SPIELER, B., AND GAETA, E. 2016. D3.3 — Analytics Dashboard: Functionalities and Usage. *EU Delivery Agreement 645215*.

COLLEGE, W. AND OF UNIVERSITY WOMEN, A. A. 1992. How schools shortchange girls: The AAUW report: a study of major findings on girls and education. *Washington, DC: AAUW Educational Foundation*.

COLLEY, A. AND COMBER, C. 2003. Age and gender differences in computer use and attitudes among secondary school students: what has changed? *Educational Research 45*, 155–165.

CONNELL, R. 1995. Masculinities. *In Polity Press*.

COOPER, A. 2003. The Origin of Personas. Retrieved February 20, 2018 from `http://www.cooper.com/journal/2008/5/the_origin_of_personas`.

COOPER, S. 2010. The design of alice. *In Trans. Comput. Educ. 10,* 15.

CORTI, K. 2006. Games-based Learning; a serious business application. *PIXELearning limited*. `www.pixelearning.com/docs/games_basedlearning_pixelearning.pdf`.

CRAIG, A., COLDWELL-NEILSON, J., AND BEEKHUYZEN, J. 2013. Are IT interventions for Girls a Special Case? *In Proceeding of the 44th ACM technical symposium on Computer science education*, 451–456.

CRESWELL, J. 2009. Research design : qualitative, quantitative, and mixed methods approache. *In Sage Publications*.

CUFF, E. 2015. The Effect and Importance of Technology in the Research Process. *In Journal of Educational Technology Systems 43,* 1, 75–97.

CUKIER, W., SHORTT, D., AND DEVINE, I. 2002. Gender and information technology: implications of definitions. *In SIGCSE Bull 34,* 4, 142–148.

CULIN, S. 1975. Games of the North American Indians. *Courier Corporation 24*.

DA ROCHA SEIXAS, L., GOMES, A., AND FILHO, I. M. 2016. Effectiveness of gamification in the engagement of students. Computers in Human Behavior. *In Computers in Human Behavior 58*, 48–63.

DAGIENE, V. AND FUTSCHEK, G. 2008. Bebras International Contest on Informatics and Computer Literacy, Criteria for Good Tasks. *ISSEP 2008, LNCS 5090, Österreichische Biber Webseite: biber.ocg.at*, 19–30.

DASGUPTA, N. AND ASAGARI, S. 2004. Seeing is believing: Exposure to counterstereotypic women leaders and its effect on the malleability of automatic gender stereotyping. *In Journal of Experimental Social Psychology 40,* 5, 642–658.

DASGUPTA, N. AND STOUT, J. 2012. Contemporary discrimination in the lab and real world: Benefits and obstancles of full-cycle psychology. *In Journal of Social Issues 60*, 817–830.

DASGUPTA, N. AND STOUT, J. G. 2014. Girls and Women in Science, Technology, Engineering, and Mathematics. *In Policy Insights from the Behavioral and Brain Sciences 1,* 1, 21–29.

DAVIES, A., KLAWE, M., NYHUS, L. C., , AND SULLIVAN, H. 2014. Gender issues in computer science education. *Vancouver, BC, Canada: Department of Computer Science, University of British Columbia*.

DAVIES, R. AND SEDLEY, L. 2009. Agile Coaching. *The Pragmatic Bookshelf*.

DAVIS, F. 1989. Perceived usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *In MIS Quarterly 13,* 3, 319–340.

DE HOYOS, M. AND BARNES, S. A. 2012. Analysing Interview Data. Warwick Institute for Employment Research. Retrieved February 18, 2018 from `https://warwick.ac.uk/fac/cross_fac/esrcdtc/coretrainingmodules/quals/analysing_interview_data_1_-_w6.pdf`.

DEE, H. M., PETRIE, K. E., BOYLE, R. D., AND PAU, R. 2009. Why are we still here? experiences of successful women in computing. *In SIGCSE Bull 41,* 3, 233–237.

DEEN, M., CHATHAM, A., AND BERNHAUPT, R. 2014. Game Jam [4Reseatch]. In *In CHI '14 Extended Abstracts on Human Factors in Computing Systems*. 25–28. http://doi.acm.org/10.1145/2559206.2559225.

DEFY MEDIA. 2017. Acumen Report: Youth Video Diet. Retrieved February 20, 2018 from `http://defymedia.com/wp-content/uploads/2017/11/Acumen_DL_booklet_16_12_04.pdf`.

DEMARCO-BROWN, D. 2013. Agile User Experience Design - A Practitioner's Guide to making it work. *In Elsevier Inc*.

DENNER, J., LYON, L., AND WERNER, L. 2015. Does Gender Matter? Women Talk about Being Female in College Computing Classes. *In Proceedings of the Third Conference on GenderIT (GenderIT '15)*, 44–48.

DENNER, J., WERNER, L., BEAN, S., AND CAMPE, S. 2005. The Girls Creating Games Program: Strategies for engaging middle school girls in information technology. *In Frontiers: A Journal of Women's Studies 26,* 1, 90–98.

DETERDING, S. 2012. Gamification: designing for motivation. *In Interactions 19,* 4, 14–17.

DETERDING, S., DIXON, D., KHALED, R., AND NACKE, L. 2011. From Game Design Elements to Gamefulness: Defining "Gamification". *In Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, 9–15.

DIENDORFER, G. 2010. Internet competence of student, types of activities, subject interests and research behavior in the 8th grade. Study commissioned by the BMUKK, dt. Internetkompetenz von SchülerInnen, Aktivitätstypen, Themeninteressen und Rechercheverhalten in der 8. Schulstufe. Studie im Auftrag des BMUKK. Retrieved February 08, 2018 from `www.demokratiezentrum.org/fileadmin/media/pdf/JIKS_Report_final.pdf`.

DOMÍNGUEZ, A., DE NAVARRETE, J. S., DE MARCOS, L., FERNÁNDEZ-SANZ, L., AND PAGÉS, C. 2013. Gamifying learning experiences: Practical implications and outcomes. *In Educational Psychology Review Computers & Education 63*, 380–392.

DONDLINGER, M. J. 2007. Educational Video Game Design: A Review of the Literature. *In Journal of Applied Educational Technology Journal of Applied Educational Technology 4,* 1.

DORMANS, J. 2012. Engineering emergence: applied theory of game design. *Universiteit van Amsterdam*.

DORNMAYR, H. AND WINKLER, B. 2016. Befragung Österreichischer LehrabsolventInnen zwei Jahre nach LehrabschlussTeilbericht im Rahmen der ibw-Studie - Hintergrundanalyse zur Wirksamkeit der betrieblichen Lehrstellenfáderung (gemäß Paragraph 19c BAG). *ibw-Studie*.

DOWNES, S. 2007. What Connectivism Is? Retrieved February 08, 2018 from `http://halfanhour.blogspot.no/2007/02/what-connectivism-is.html`.

EAGLY, A., GARTZIA, L., AND CARLI, L. 2014. Female Advantage: Revisited. *In S. Kumra, Simpson R. & R- Birle (Eds.), The Oxford Handbook of Gender in Organizations. Oxford University Press*.

EAGLY, A. AND WOOD, W. 2012. Social role theory. *In P.A.M. Van Lange, A.W. Kruglanski, & E.T. Higgins (Eds.), Handbook of theories of social psychology*, 458–476.

EBERHARDT, R. 2016. No One Way to Jam: Game Jams for Creativity, Learning, Entertainment, and Research. *In Proceedings of the International Conference on Game Jams, Hackathons, and Game Creation Events*, 34–37.

ECCLES, J. 2007. Where are all the women? gender differences in participation in physical science and engineering. *In S.J., Ceci and W.M., Williams (Eds.), Why Aren't More Women in Science? Washington DC: American Psychological Association*, 199–210.

ECCLES, J. 2011. Gendered educational and occupational choices: Applying the Eccles et al.1 model of achievement-related choices. *In International Journal of Behavioral Development 35,* 3, 195–201.

ECKSTEIN, K. 2014. Zahlen, Fakten, Analysen. Chancengleichheit an der Uni Graz. Retrieved March 20, 2018 from `https://static.uni-graz.at/fileadmin/Koordination-Gender/Services/Zahlen_Fakten_Analysen_UniGraz2014.pdf`. Koordinationsstelle für Geschlechterstudien, Frauenforschung und Frauenförderung der Karl-Franzens-Universit Graz.

EL-NASR, M. S., YUCEL, I., ZUPKO, J., ANDREA, AND SMIT, T. B. 2007. Middle-to-High School Girls as Game Designers — What are the Implications? *Academic Days '07*.

EMARKETER. 2015. Multiscreening Coexists Alongside TV in Western Europe-Increased device adoption drives multiscreening during TV time. Retrieved December 09, 2017 from `http://www.emarketer.com/Article/Multiscreening-Coexists-Alongside-TV-Western-Europe/10`.

ENGLER, S. AND WIELAND, H. F. 1995. Ent-Dramatisierung der Differenzen. Studentinnen und Studenten der Technikwissenschaften. *Bielefeld: Kleine Verlag*.

ENTERTAINMENT SOFTWARE ASSOCIATION. 2016. Essential facts about the computer and video game industry. Retrieved January 18, 2018 from `http://essentialfacts.theesa.com/Essential-Facts-2016.pdf`.

ESPINOSA, L. L. 2015. Where Are the Women in STEM? Retrieved September, 10, 2017 from `http://higheredtoday.org/2015/03/03/where-are-the-women-in-stem/`.

EUROPE OF COUNCIL. 1998. Gender mainstreaming. Retrieved February 20, 2018 from `https://www.coe.int/en/web/genderequality/gender-mainstreaming`.

EUROPEAN COMMISSION [1]. 2016. A NEW SKILLS AGENDA FOR EUROPE. Working together to strengthen human apital, employability and competitiveness. Retrieved Janury 18, 2018 from `http://ec.europa.eu/social/main.jsp?catId=1223`.

EUROPEAN COMMISSION [2]. 2016. Women in digital — a gap and an opportunity. Retrieved December 8, 2017 from https://ec.europa.eu/digital-single-market/en/blog/women-digital-gap-and-opportunity.

EUROPEAN COMMISSION [3]. 2013. Women on Boards: Share of Women up to 16.6% as European Parliament Committees back Commission proposal. Retrieved March 20, 2018 from http://europa.eu/rapid/press-release_IP-13-943_en.htm.

EUROPEAN SCHOOLNET. 2015. Computing our future. Computer programming and coding - Priorities, school curricula and initiatives across Europe. Retrieved January 5, 2018 http://fcl.eun.org/documents/10180/14689/Computing+our+future_final.pdf/746e36b1-e1a6-4bf1-8105-ea27c0d2bbe0.

EUROPEAN STATISTICS. 2018. Women in science and technology. Retrieved March 20, 2018 from http://ec.europa.eu/eurostat/web/products-eurostat-news/.

EUROSTAT. 2017. ICT specialists in employment. Retrieved March 20, 2018 from http://ec.europa.eu/eurostat/statistics-explained/index.php/ICT_specialists_in_employment.

FEDERAL CHANCELLERY AUSTRIA. 2009. Bundeskanzleramt Österreich: Rechtliche und politische Grundlagen der Gleichstellung von Frauen und Männern in Österreich. Retrieved February 26, 2018 from http://www.imag-gmb.at/cms/imag/attachments/1/9/7/CH0619/CMS1465981568160/rechtliche_grundlagen_gleichstellung.pdf. Gender Mainstreaming. No. 7. Issue 2.

FEDERAL MINISTRY OF EDUCATION AUSTRIA [1]. 2017. Highscool Curricula (BMB - Bundesministerium für Bildung: Lehrpläne der AHS Oberstufe). Retrieved February 08, 2018 from https://www.bmb.gv.at/schulen/unterricht/lp/lp_ahs_oberstufe.html.

FEDERAL MINISTRY OF EDUCATION AUSTRIA [2]. 2017. Highschool Curricula - Informatics mandatory 9th grade, (BMB - Bundesministerium für Bildung: Lehrpläne der AHS Oberstufe - Informatik 5. Klasse). Retrieved February 08, 2018 from https://www.bmb.gv.at/schulen/unterricht/lp/lp_neu_ahs_14_11866.pdf.Adrian.

FEDERAL MINISTRY OF EDUCATION AUSTRIA [3]. 2017. Highschool Curricula - Informatics optional 10th-12th grade, (BMB - Bundesministerium für Bildung: Lehrpläne der AHS Oberstufe - Informatik 6.-8. Klasse). Retrieved February 08, 2018 from https://www.bmb.gv.at/schulen/unterricht/lp/lp_neu_ahs_14_11866.pdf.

FEDERAL MINISTRY OF EDUCATION AUSTRIA [4]. 2017. Education in Austria 2016/2017. Retrieved February 10, 2018 from https://www.bmb.gv.at/enfr/school/bw_en/bildungswege2016_eng.pdf?5te5kh.

FEDERAL MINISTRY OF EDUCATION AUSTRIA [5]. 2017. Mandatory exercise "Digital literacy" in secondary education 1, Content for piloting in the school year 2017/18. Retrieved February 07, 2018 from https://tinyurl.com/y78wov7a.

FEINDLER, E. AND LIEBMAN, M. 2015. Behavioral Assessment in School Settings. *In Cognitive and Behavioral Interventions in the Schools. Springer*.

FERDIG, R. AND WINN, B. 2009. The Design, Play, and Experience Framework. *In book Handbook of Research on Effective Electronic Gaming in Education*.

FERRANT, G., PESANDO, L. M., AND NOWACKA, K. 2014. Unpaid Care Work: The missing link in the analysis of gender gaps in labour outcomes. Retrieved February 20, 2018 from https://www.oecd.org/dev/development-gender/Unpaid_care_work.pdf. OECD Development Centre.

FERREIRA, A., PEREIRA, E., ANACLETO, J., CARVALHO, A., AND CARELLI, I. 2008. The common sense-based educational quiz game framework "What is it?". *In ACM International Conference Proceeding Series 378,* 3, 338–339.

FIELDS, D. A., GIANG, M., AND KAFAI, Y. 2014. Programming in the wild: trends in youth computational participation in the online scratch community. *In Proceedings of the 9th Workshop in Primary and Secondary Computing Education*, 2–11.

FILSECKER, M. AND KERRES, M. 2014. Engagement as a volitional construct: a framework for evidence-based research on educational games. *In Simulation & Gaming 45*, 450–470.

FISCUTEAN, A. 2017. Women in tech: Why Bulgaria and Romania are leading in software engineering. Retrieved March 20, 2018 from http://www.zdnet.com/article/women-in-tech-why-bulgaria-and-romania-are-leading-in-software-engineering/. ZDNet by CBS Interactive.

FISHER, A. AND MARGOLIS, J. 2002. Unlocking the clubhouse: The Carnegie Mellon experience. *In SIGCSE Bull 34,* 2, 37–41.

FISHER, A., MARGOLIS, J., AND MILLER, F. 2015. Are Females Disinclined to Tinker in Computer Science? *In Proceedings of the twenty-eighth SIGCSE technical symposium on Computer science education*, 106–110.

FLABBI, L., MACIS, M., MORO, A., AND SCHIVARDI, F. 2014. Do Female Executives Make a Difference? The Impact of Female Leadership on Gender Gaps and Firm Performance. Retrieved December 09, 2017 from http://ftp.iza.org/dp8602.pdf. Discussion Paper No. 8602.

FORD, J. L. 2009. Scratch programming for Teens. *In Computer Science Books*.

FORMANOWICZ, M., CISLAK, A., HORVATH, L., AND SCZESNY, S. 2015. Capturing Socially Motivated Linguistic Change. Different effects of gender-fair language on support for social initiatives in Austria and Poland. *In Frontiers in Psychology 6,* 1617.

FOWLER, A., KHOSMOOD, F., AND ARYA, A. 2013. The Evolution and Significance of the Global Game Jam. *In Workshop Proceedings of the 8th International Conference on the Foundations of Digital Games*.

FRAILLON, J., SCHULZ, W., AND AINLEY, J. 2013. International Computer and Information Literacy Study: Assessment Framework. *In ICT — Digital Literacy*. https://research.acer.edu.au/ict_literacy/9.

FRASER, J., L. ATKINS, L., AND HALL, R. 2013. DigiLit Leicester: Initial Project Report 2013. *In Leicester: Leicester City Council*.

FREITAS, N., FILHO, D., AND BARBOSA, E. F. 2013. A requirements catalog for mobile learning environments. *In Proceedings of the 28th Annual ACM Symposium on Applied Computing*, 1266–1271.

FRENKEL, L. 2018. Online courses don't work, but education can still be disrupted. Retrieved Feburary 1, 2018 from http://www.chinadaily.com.cn/a/201801/12/WS5a5851a0a3102c394518edcc.html.

FRIEZE, C. AND QUESENBERRY, J. 2015. *Kicking Butt in Computer Science: Women in Computing at Carnegie Mellon University*. Dog Ear Publishing.

FUNKE, A., GELDREICH, K., AND HUBWIESER, P. 2017. Analysis of scratch projects of an introductory programming course for primary school students. *In Global Engineering Education Conference (EDUCON).*

GABAY-EGOZI, L., SHAVIT, Y., AND YAISH, M. 2015. Gender Differences in Fields of Study: The Role of Significant Others and Rational Choice Motivations. *In European Sociological Review 31,* 3, 284–297.

GAETA, E. AND CEA, G. 2017. D5.5 — Report and findings from experimental pilot in Spain. *EU Delivery Agreement 645215.*

GALAS, C. AND FREUDENBERG, R. 2010. Learning with Squeak Etoys. *In Constructionism 2010.*

GALDI, S., CADINU, M., AND TOMASETTO, C. 2014. The roots of stereotype threat: when automatic associations disrupt girls' math performance. *In Child Development 85,* 1.

GAMES, A. AND KANE, L. 2011. Exploring adolescent's STEM learning through scaffolded game design. *In Proceedings of the 6th International Conference on Foundations of Digital Games*, 1–8. New York, USA.

GARCÍA-PEÑALVO, F. J., REES, A. M., HUGHES, J., JORMANAINEN, I., TOIVONEN, T., AND VERMEERSCH, J. 2016. A survey of resources for introducing coding into schools. *In Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality*, 19–26.

GARCZYNSKI, J. 2014. Chi-Square Test in SPSS (PASW). *Reference Department. Albert S. Cook Library.*

GARDNER, H. 1983. *Frames of Mind.* In Basic Book Inc.

GARDNER, H. 1991. *The unschooled mind: how children think and how schools should teach.* In Basic Book Inc.

GEORGIOS, F. AND KIRIAKI, S. 2009. Influence of the Familiarization with "Scratch" on Future Teachers' Opinions and Attitudes about Programming and ICT in Education. *In Annual Joint Conference Integrating Technology into Computer Science Education*, 258–262.

GERMAN FEDERAL STATISTICAL OFFICE. 2017. Zahl der Studierenden steigt im Wintersemester 2017/2018 erneut an. Retrieved February 04, 2018 from `https://www.destatis.de/DE/PresseService/Presse/Pressemitteilungen/2017/11/PD17_427_213.html`.

GHARIBYAN, H. AND GUNSAULUS, S. 2006. Gender gap in computer science does not exist in one former Soviet Republic: Results of a study. *In Proceedings of the 11th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, 222–226.

GIANNAKOS, M. N., JACCHERI, L., AND LEFTHERIOTIS, I. 2014. Happy Girls Engaging with Technology: Assessing Emotions and Engagement Related to Programming Activities. *In Learning and Collaboration Technologies. Designing and Developing Novel Learning Experiences 8523*, 398–409.

GIBBONS, T. 2013. COR: A new course framework based on elements of game design. *In W.D. Armitage (Ed.), Proceedings of 14th Annual Conference in Information Technology Education*, 77–82.

GIBBS, R. W. AND MUELLER, R. A. 1990. Conversation as coordinated, cooperative interaction. *In Cognition, computing, and cooperation*, 95–104.

GIBBS, S. 2014. Women in technology: no progress on inequality for 10 year. Retrieved March 20, 2018 from https://www.theguardian.com/technology/2014/may/14/women-technology-inequality-10-years-female. The Guardian.

GILDEMEISTER, R. 2010. Doing Gender. *In Becker R., Kortendiek B. (eds) Handbuch Frauen- und Geschlechterforschung*.

GILTEMEISTER, R. AND ROBERT, G. 2008. Geschlechterdifferenzierung in lebenszeitlicher Perspektive. *Interaktionen — Institute — Biografie. Wiesbaden — VS Verlag für Sozialwissenschaften*.

GLEICHBEHANDLUNGSANWALTSCHAFT ÖSTERREICH . 2011. Geschlechtergerechte Sprache. Retrieved February 10, 2018 from https://www.gleichbehandlungsanwaltschaft.gv.at/.

GLOBAL GAMES MARKET REPORT. 2016. The Global Games Market Reaches $99.6 Billion in 2016, Mobile Generat-ing 37%. Retrieved Feburary 06, 2018 from https://newzoo.com/insights/articles/global-games-market-reaches-99-6-billion-2016-mobile-generating-37/.

GLORIA, A. D., BELLOTTI, F., AND BERTA, R. 2014. Serious Games for education and training. *In CC BY-NC-ND 4.0*.

GLOVER, I. 2013. Play As You Learn: Gamification as a Technique for Motivating Learners. *World Conference on Educational Multimedia, Hypermedia and Telecommunications*.

GOBIERNO DE ESPANA MINISTERIO DE EDUCACIO. 2009. Spanish Education System. Retrieved February 01, 2018 from http://www.mecd.gob.es/educacion-mecd/dms/mecd/educacion-mecd/mc/redie-eurydice/estudios-informes/redie/informes-generales/spanish-education-2009.pdf.

GODDARD, W., BYRNE, R., AND MUELLER, F. 2014. Playful Game Jams: Guidelines for Designed Outcomes. In *In Proceedings of the 2014 Conference on Interactive Entertainment*. 1–10.

GOOD, C., RATTAN, A., AND DWECK, C. 2012. Why do women opt out? Sense of belonging and women's representation in mathematics. *In Journal of Personality and Social Psychology 102*, 700–717.

GOODE, J., CHAPMAN, G., AND MARGOLIS, J. 2012. Beyond curriculum: the exploring computer science program. *In Magazine ACM Inroads 3,* 1, 47–53.

GOODE, J., ESTRELLA, R., AND MARGOLIS, J. 2006. Lost in translation: Gender and high school computer science. *In Aspray, W. & Cohoon, J.M. (Eds.) Women in IT: Reasons on the Reasons of Under-Representation. Cambridge, MA: MIT Press*.

GOOGLE. 2016. TNS Infratest; Google; Connected Consumer Survey 2016. Retrieved March 17, 2018 from https://www.consumerbarometer.com/en/graph-builder/?question=M1&filter=country:austria.

GOOGLE. 2018. Run A/B tests on your Store Listing. Retrieved February 20, 2018 from https://support.google.com/googleplay/android-developer/answer/6227309?hl=en-GB.

GOOGLE. 2018. Set up an alpha, beta, or internal test. Retrieved February 08, 2018 from https://support.google.com/googleplay/android-developer/answer/3131213?hl.

GOOGLE DEVELOPER. 2018. Use Store List Entries to convert more visits to installations. Retrieved February 20, 2018 from https://developer.android.com/distribute/best-practices/grow/store-listing-experiments.html.

GOOGLE DEVELOPERS. 2018. Multiple APK Support. Retrieved February 20, 2018 from https://developer.android.com/google/play/publishing/multiple-apks.html.

GOOGLE INC. 2017. The world of women and mobile gaming. Retrieved March 20, 2018 from http://services.google.com/fh/files/misc/changethegame_white_paper.pdf.

GOOGLE INC. 2018. Change the Game. Retrieved March 20, 2018 from https://play.google.com/about/changethegame.

GORRIZ, C. M. AND MEDINA, C. 2000. Engaging girls with computers through software games. *In Commun. ACM 43 43,* 1, 42–49.

GOV.UK DEPARTMENT FOR EDUCATION. 2013. National curriculum in England: computing programmes of study. Retrieved Feburary 18, 2018 from https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study.

GRANDL, M. AND EBNER, M. 2017. Informatische Grundbildung - ein Ländervergleich. Retrieved February 08, 2018 from http://www.medienimpulse.at/pdf/Medienimpulse_Informatische_Grundbildung_____ein_Laendervergleich_Grandl_20170514.pdf.

GRANDL, M., HÖLLERBAUER, B., EBNER, M., AND SLANY, W. 2017. Ein offenes Unterrichtskonzept für den Einstieg in die Programmierung mit Hilfe von "Pocket Code". Coding als Baustein der digitalen Grundbildung. *Schule Aktiv (Sonderheft des BMB)*, 32–37.

GRANT, J., HOORENS, S., SIVADASAN, S., LOO, M., AND DAVANZO, J. 2004. Low Fertility and Population Ageing. Causes, Consequences, and Policy Options. Retrieved February 08, 2018 from https://www.rand.org/content/dam/rand/pubs/monographs/2004/RAND_MG206.pdf.

GREENBERGER, M. 1962. Computers in the World of the Future. *In Cambridge, MA: MIT Press*.

GRIGOREANU, V., CAO, J., KULESZA, T., BOGART, C., RECTOR, K., BURNETT, M., AND WIEDENBECK, S. 2008. Can feature design reduce the gender gap in end-user software development environments? *In IEEE Symp. Visual Lang. Human-Centric Computing*, 149–156.

GROSSARTH-MATICEK, R. 2002. Selbstregulation, Autonomie und Gesundheit. Krankheitsrisiken und soziale Gesundheitsressourcen im sozio-psycho-biologischen System. *Kapitel 4. Grossarthsche Verhaltenstypologie*.

GRÜNBERG, L. 2011. *From Gender Studies to Gender In Studies - Case Studies of Gender-Inclusive Curriculum in High Education*.

HAINES, L. 2004. Why are there so few women in games? *In Digital Games Research Conference, Changing Views: Worlds in Play*, 1–23. Vancouver.

HAMARI, J. AND KOIVISTO, J. 2013. Social Motivations To Use Gamification: An Empirical Study Of Gamifying Exercise. *In ECIS 2013 Completed Research*.

HARFORD, T. 2017. Grace Hopper's compiler: Computing's hidden hero. Retrieved March 20, 2018 from http://www.bbc.com/news/business-38677721. BBC World Service, 50 Things That Made the Modern Economy.

HARVEY, B. AND MÖNING, J. 2010. Bringing 'No Ceiling' to Scratch: Can One Language Serve Kids and Computer Scientists? *In Constructionism 2010*.

HASLER. 2013. Informatik im Lehrplan 21. Ein grundsätzlicher Positionsbezug zum Wohl und Nutzen des Denk- und Werkplatzes Schweiz. Retrieved Feburar 6, 2018 from `http://fit-in-it.ch/sites/default/files/downloads/dok_2013-06-20_informatik_im_lehrplan_21.pdf`.

HEETER, C., CHU, K., EGIDIO, R., AND MISHRA, P. 2000. Do Girls Prefer Games Designed by Girls? . *In Proceedings from ISA:55 the Annual Conference of the International Communication Association*, 1–31. NY.

HEILMAN, M. 1983. Sex bias in work settings: The Lack of Fit model. *In Research in Organizational Behavior 5*, 269–298.

HEILMAN, M. 2012. Gender stereotypes and workplace bias. Research. *In Organizational Behavior 32*, 113–135.

HENNINK, M. M. 2014. Focus group discussions. Understanding Qualitative Research. *In Oxford University Press*.

HENTSCHE, T. AND HORVATH, L. 2015. Passende Talente ansprechen - Rekrutierung und Gestaltung von Stellenanzeigen. In C. Peus, S. Braun, T. Hentschel & D. Frey (Eds.), Personalauswahl in der Wissenschaft - Evidenzbasierte Methoden und Impulse für die Praxis. *Heidelberg: Springer* 65-82.

HERCY, N., WINSTON, M., CALVIN, C., AND TAK-WAI, C. 2009. Equal opportunity tactic: Redesigning and applying competition games in classrooms. *In Computers & Education 53,* 3, 866–876.

HEWLETT, S. A. AND SHERBIN, L. 2014. Athena Factor 2.0: Accelerating Female Talent in Science, Engineering & Technology. *In Center for Talent Innovation*. `http://www.talentinnovation.org/assets/Athena-2-ExecSummFINAL-CTI.pdf`.

HIDI, S. AND HARACKIEWICZY, J. 2000. Motivating the Academically Unmotivated: A Critical Issue for the 21st Century. *In Review of Educational Research 70,* 2, 151–179.

HIRSCHFELDER, D. 2006. Der "Nürnberger Trichter" — Ein Allheilmittel gegen die Dummheit? *In KulturGUT — Aus der Forschung des Germanischen Nationalmuseums 8*, 3–5.

HOLLIDAY, M. AND LUGINBUHL, D. 2004. Peer-centered service learning. *In Proceedings of the 34th Frontiers in Education*, 1–6.

HORVATH, L. 2015. Gender-fair language in the context of recruiting and evaluating leaders. *In I.M. Welpe, P. Brosi, L. Ritzenhöfer, & T. Schwarzmüller (Eds.), Auswahl und Beurteilung von Frauen und Männern als Führungskräfte in der Wirtschaft — Herausforderungen, Chancen und LÃűsungen*, 263–272.

HORVATH, L. K., MERKEL, E. F., MAASS, A., AND SCZESNY, S. 2016. Does Gender-Fair Language Pay Off? The Social Perception of Professions from a Cross-Linguistic Perspective. *In Front Psychol. 6*.

HUBWIESER, P. 2007. Didaktik der Informatik: Grundlagen, Konzepte, Beispiele. *eXamen.press*.

HUFF, C. 2002. Gender, software design, and occupational equity. *In SIGCSE Bull 34,* 2, 112–115.

HUGHES-ROBERTS, T., BURTON, A., MARINOVS, D., AND TINNEY, J. 2017. D5.3 — Report and findings from experimental pilot in the United Kingdom. *EU Delivery Agreement 645215*.

HUIZINGA, J. 2009. Homo Ludens: A Study of the Play-Element in Culture. *In Paperback first edition 1946*.

HULSEY, C., PENCE, T. B., AND HODGES, L. F. 2014. Camp CyberGirls. *In Proceedings of the 45th ACM technical symposium on Computer science education*, 331–336.

HUNICKE, R., LEBLANC, M., AND ZUBEK, R. 2004. MDA: A formal approach to game design and game research. *In Proceedings of the AAAI Workshop on Challenges in Game AI 4*.

HUNT, V., LAYTON, D., AND PRINCEC, S. 2015. Why diversity matters. Retrieved October 18, 2017 from `https://www.mckinsey.com/business-functions/organization/our-insights/why-diversity-matters/`.

IACONO, S. 2013. Background on NSF Broadening Participation. Retrieved March 20, 2018 from `http://www.nsf.gov/od/broadeningparticipation/bp.jsp`. National Science Foundation.

IBÁÑEZ, M. B., DI-SERIO, A., AND DELGADO-KLOOS, C. 2014. Gamification for engaging computer science students in learning activities: A case study. *In IEEE Transactions n Learning Technologies 7,* 3, 291–301.

ILDIKÓ TASNÁDI, L. C. AND FARKAS, K. 2016. Teaching programming constructively and playfully. *In Proceedings of Constructionism*, 109–116.

INFORMATICS EUROPE/ACM EUROPE. 2016. Informatics Education in Europe: Are We All In The Same Boat? *Report by The Committee on European Computing Education (CECE) Jointly established by Informatics Europe & ACM Europe*. Retrieved March 03, 2018 `http://www.informatics-europe.org/news/382-informatics-education-in-europe-are-we-on-the-same-boat.html`.

IT MANAGER DAILY. 2018. Women in Technology: Triumphs and Barriers. Retrieved March 20, 2018 from `http://www.itmanagerdaily.com/women-in-technology-infographic/`.

JAFFA, V. 2016. In a Jam Between Community and Capitalism: A Critical Look at Game Jams. Retrieved October 18, 2017 from `https://modelviewculture.com/news/in-a-jam-between-community-and-capitalism-a-critical-look-at-game-jams`.

JAIME, S. AND RUBY, O. 2011. Problem solving and collaboration using mobile serious games. *In Elsevier Ltd*.

JAMES, W., MAUREEN, D., RUDY, G., AND ZACHARY, H. 2013. An Informatics Perspective on Computational Thinking. In In Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education. *In Proceedings of the Third International Conference on Creating, Connecting and Collaborating through Computing*, 4–9.

JÁRVINEN, A. 2007. Introducing Applied Ludology: Hands-on Methods for Game Studies. *In Situated Play: DiGRA 2007 Conference Proceedings*.

JARZ, T. 2016. Current problems in school informatics. University of Applied Sciences Styria (dt. Aktuelle Probleme in der Schulinformatik. Pädagogische Hochschule Steiermark.). Retrieved February 8, 2018 from `https://www.ahs-informatik.com/tagungsband-25-jahre-schulinformatik/informatik-ein-fach/`. p.116-120.

JAYANTH, M. 2014. 52% of gamers are women - but the industry doesn't know it. Retrieved October, 16, 2017 from `https://www.theguardian.com/commentisfree/2014/sep/18/52-percent-people-playing-games-women-industry-doesnt-know`.

JENSON, J., CASTELL, S., AND FISHER, S. 2007. Girls playing games: rethinking stereotypes. *In Proceedings of the 2007 conference on Future Play*, 9–16.

JESPER, J. 2007. A Certain Level of Abstraction. *In Situated Play: DiGRA 2007 Conference Proceedings*.

JINGLI, S. 2017. China's mobile games market posts $15b revenue in 2017. Retrieved February 08, 2018 http://www.chinadaily.com.cn/a/201801/12/WS5a5851a0a3102c394518edcc.html.

JOHNSON, C., MCGILL, M., BOUCHARD, D., BRADSHAW, M. K., BUCHELI, V. A., MERKLE, L. D., SCOTT, M. J., SWEEDYK, Z., ANGEL, J., XIAO, Z., AND ZHANG, M. 2016. Game Development for Computer Science Education. *In Proceedings of the 2016 ITiCSE Working Group Reports (ITiCSE '16).*, 23–44.

JONES, M. G., BRADER-ARAJE, L., LISA WILSON CARBONI, CARTER, G., RUA, M. J., BANILOWER, E., AND HATCH, H. 2000. Tool time: Gender and students' use of tools, control, and authority. *In J. Res. Sci. Teach. 37,* 8, 760–783.

JONSSON, J. 1999. Explaining gender differences in educational choice: an empirical assessment of a rational choice model. *In European Sociological Review 15*, 391–404.

JULIZAERMA, M. AND SORI, Z. 2012. Gender Diversity in the Boardroom and Firm Performance of Malaysia. *In Public Listed Companies. Procedia — Social and Behavioral Science 65*, 1077–1085.

JULKUNEN, J. 2015. The Future of Match 3 — What You Need to Know — PART I. Retrieved February 26, 2018 from http://www.gamerefinery.com/the-future-of-match-3-what-you-need-to-know-part-i/. GameRefinery.

KABATOVA, M., KALAS, I., AND TOMCSANYIOVA, M. 2016. Programming in Slovak Primary Schools. *In Olympiads in Informatics 10*, 125–159.

KAFAI, Y. 2006. Playing and making games for learning: Instructionist and constructionist perspectives for game studies. *In Games & Culture*.

KAFAI, Y. 2008. Considering Gender in Digital Games: Implications for Serious Game Designs in the Learning Sciences. *In Proceedings of the 8th international conference on International conference for the learning sciences*, 422–429.

KAFAI, Y. AND BURKE, Q. 2013. Computer programming goes back to school. *In Phi Delta Kappan 95,* 1, 61.

KAFAI, Y., FIELDS, D., ROQUE, R., BURKE, W., AND MONROY-HERNÁNDEZ, A. 2012. Collaborative agency in youth online and offline creative production in Scratch. *In Research and Practice in Technology Enhanced Learning 7,* 2, 63–87.

KAFAI, Y. AND RESNICK, M. 1996. Constructionism in Practice: Designing, Thinking, and Learning in a Digital World. *Mahwah, New Jersey, Lawrence Erlbaum*.

KAFAI, Y., SEARLE, K. A., AND FIELDS, D. A. 2014. A Crafts-Oriented Approach to Computing in High School: Introducing Computational Concepts, Practices and Perspectives with E-Textiles. *In Transactions on Computing Education 14,* 1, 1–20.

KAFAI, Y. AND VASUDEVAN, V. 2015. Hi-Lo tech games: crafting, coding and collaboration of augmented board games by high school youth. *In Proceedings of the 14th International Conference on Interaction Design and Children*, 130–139.

KAHN, K. 2017. A half-century perspective on Computational Thinking. *In technologias, sociedade e conhecimento 4,* 1.

KAITILA, C. 2012. *The Game Jam Survival Guide*. In PACKT Publishing.

KANGAS, M. 2010. Creative and playful learning: Learning through game co-creation and games in a playful learning environment. *In Thinking Skills and Creativity 5,* 1, 1–15.

KAPP, K. 2012. The gamification of learning and instruction: Game-based methods and strategies for training and education. *San Francisco, CA: John Wiley & Sons.*.

KAY, A., ROSE, K., INGALLS, D., KAEHLE, T., MALONEY, J., AND WALLACE, S. 1997. Etoys & SimStories. In: ImagiLearning Group. *Walt Disney Imagineering*.

KELLY, J. 2012. Learning Theories. Retrieved February 08, 2018 http://thepeakperformancecenter.com/educational-learning/learning/theories/. The Peak Performance Center.

KERR, A. 2006. The Business of Making Games. In *Understanding Digital Games*. Sage Publications Ltd.

KETELAARS, E. 2017. What Place for Gender Mainstreaming in the EU's Framework on Support to Transitional Justice? *In European Foreign Affairs Review 22,* 3, 323–340.

KHALEEL, F. L., ASHAARI, N. S., MERIAM, T. S., WOOK, T., AND ISMAIL, A. 2015. The Study of Gamification Application Architecture for Programming Language Course. *In Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication*.

KHAN, N. Z. AND LUXTON-REILLY, A. 2016. Is computing for social good the solution to closing the gender gap in computer science? *In Proceedings of the Australasian Computer Science Week Multiconference*.

KHAZAN, O. 2018. The More Gender Equality, the Fewer Women in STEM. Retrieved March 20, 2018 from https://www.theatlantic.com/science/archive/2018/02/the-more-gender-equality-the-fewer-women-in-stem/553592/. The Atlantic.

KISH-GEPHART, J., HARRISON, D., AND TREVINO, L. 2010. Bad apples, bad cases, and bad barrels: meta-analytic evidence about sources of unethical decisions at work. *In The Journal of applied psychology 95,* 1, 1–31.

KO, A. J. AND DAVIS, K. 2017. Computing Mentorship in a Software Boomtown: Relationships to Adolescent Interest and Beliefs. *In Proceedings of the 2017 ACM Conference on International Computing Education Research*, 236–244.

KOFFLER, T. 2018. TU Graz - Facts & Figures 2016/17. Retrieved March 20, 2018 from https://www.tugraz.at/fileadmin/user_upload/tugrazInternal/TU_Graz/Universitaet/TU_Graz_kompakt/Facts_Figures_2016_2017.pdf. Holder & Editor: Technische Universität Graz.

KOH, K., BASAWAPATNA, A., BENNETT, V., AND REPENNING, A. 2010. Towards the Automatic Recognition of Computational Thinking for Adaptive Visual Language Learning. *In Proceedings of the 2010 IEEE Symposium on Visual Languages and Human-Centric Computing, IEEE Computer Society*, 59–66.

KÖLTZSCH, T. 2017. Die finnische Modellschule hat Tablets statt Schreibtische. Retrieved January 08, 2018 https://www.golem.de/news/

it-in-der-schule-die-finnische-modellschule-hat-tablets-statt-schreibtische-1712-131673.html.

KONDRAT, X. 2015. Gender and video games: How is female gender generally represented in various genres of video games? *In Journal of Comparative Research in Athoropology and Sociology 6,* 1. http://compaso.eu.

KONZACK, L. 2002. Computer Game Criticism: A Method for Computer Game Analysis. *In CGDC Conference Proceedings, Frans Mayra, ed., Tampere University Press*, 89–100.

KOP, R. AND HILL, A. 2008. Connectivism: Learning theory of the future or vestige of the past? *In The International Review of Research in Open and Distance Learning 9,* 3.

KOPCHA, T., DING, L., AND NEUMANN, K. 2016. Teaching Technology Integration to K-12 Educators: A 'Gamified' Approach. *In TechTrends 60,* 1, 62–69.

KRETZSCHMAR, R. 2016. Informatik für alle in der Schweiz. *In OCG Journal*, 32–33.

KRIEGER, S., ALLEN, M., AND RAWN, C. 2015. Are feamles disinclined to tinker in computer science? *In Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, 102–107.

KROOK, A. 2018. Tech and the Gender Pay Gap. Retrieved March 20, 2018 from https://www.payscale.com/data-packages/gender-pay-gap/women-in-tech.

KRUEGER, R. 1994. Focus groups: A practical guide for applied research. *In Sage Publications, Inc.*.

KUMAR, D. 2011. Ready for a third peak? *In ACM Inroads 2,* 3, 10–11.

KWUS. 2016. Kids Wireless Use Facts. Retrieved February 08, 2018 from http://www.growingwireless.com/get-the-facts/quick-facts.

LADNER, R. AND ISRAEL, M. 2016. 'For All' in 'Computer Science For All. *In Communications of the ACM 9*, 26–28.

LAMBORELLE, A. AND FERNANDEZ, L. 2016. Woman in ICT - How do EU member states measure up? Retrieved December 09, 2017 http://www.euractiv.com/section/digital/infographic/women-in-ict-how-do-eu-member-states-measure-up/.

LANKOSKI, P. AND BJÖRK, S. 2015. Game Research Methods. An Overview. *In ETC Press 2015*.

LAREN, C. M. 2004. A comparison of student persistence and performance in online and classroom business statistics experiences. *In Decision Sciences Journal of Innovative Education 2,* 1, 1–10.

LAW, E. E., VERMEEREN, A., HASSENZAHL, M., AND BLYTHE, M. 2017. The hedonic/pragmatic model of user experience. Towards a UX Manifesto. Retrieved January 18, 2018 from http://www.academia.edu/2880396/The_hedonic_pragmatic_model_of_user_experience.

LEE, J. H., KARLOVA, N., CLARKE, R. I., THORNTON, K., AND PERTI, A. 2014. Facet Analysis of Video Game Genres. *In Proceedings of the iConference 2014*, 125–139.

LENHART, A., KAHNE, J., MIDDAUGH, E., MACGILL, A., EVANS, C., AND VITAK, J. 2008. Teens, Video Games and Civics - Part 1.1: Who Is Playing Games? Retrieved March 20, 2018 from http://www.pewinternet.org/2008/09/16/part-1-1-who-is-playing-games.

LEWIS, C. 2007. Attitudes and beliefs about computer science among students and faculty. *In SIGCSE Bull 34,* 2, 37–41.

LEWIS, C., ESPER, S., BHATTACHARYYA, V., FA-KAJI, N., DOMINGUEZ, N., AND SCHLESINGER, A. 2014. Children's perceptions of what counts as a programming language. *In Journal of Computing Sciences in Colleges 29,* 4, 123–133.

LI, F. W. AND WATSON, C. 2011. Game-Based Concept Visualization for Learning Programming. *In Proceedings of the third international ACM workshop on Multimedia technologies for distance learning*, 37–42.

LIEBERMAN, D. 2006. What can we learn from playing interactive games? *In P. Vorderer & J. Bryant (Eds.), Playing video games: Motives, responses, and consequences. Mahwah, NJ: Lawrence Erlbaum Associates.*.

LOCKHEED, M. AND HARRIS, A. 1984. Cross-Sex Collaborative Learning in Elementary Classrooms. *In American Educational Research Journal 21,* 2, 275–294.

LOCKWOOD, P. 2006. Someone like me can be successful: do college students need same-gender role models? *In Psychology of Women Quarterly*.

LYE, S. AND KOH, J. 2014. Review on teaching and learning of computational thinking through programming: What is next for K-12? *In Computers in Human Behavior 41*, 51–61.

LYNCH, T., TOMPKINS, J. E., VAN DRIEL, I. I., AND FRITZ, N. 2016. Sexy, Strong, and Secondary: A Content Analysis of Female Characters in Video Games across 31 Years. *In Journal of Communication 66,* 4, 564–584.

MAASS, S. 2018. Das Modell. Soziotechnische Systemgestaltung & Gender. Retrieved February 18, 2018 from `http://www.informatik.uni-bremen.de/soteg/gerd/?action=modell`.

MACCONNELL, S. 1998. Software project survival guide. *In Microsoft Press*.

MANN, A. AND DIPRETE, T. 2013. Trends in gender segregation in the choice of science and engineering majors. *In Social Science Research 42,* 6, 1519–1541.

MANNILA, L., DAGIENE, V., DEMO, B., GRGURINA, N., MIROLO, C., ROLANDSSON, L., AND SETTLE, A. 2014. Computational thinking in K-9 education. *In Proceedings of the Working Group Reports of the 2014 on Innovation & Technology in Computer Science Education Conference*, 1–29.

MARGOLIS, J., FISHER, A., AND MILLER, F. 1999. Caring about connections: Gender and computing. *In IEEE Technology and Society Magazine 18,* 4, 13–20.

MARGOLIS, J., FISHER, A., AND MILLER, F. 2014. Geek Mythology. *In Women in Computer Sciences: Closing the Gender Gap in Higher Education*. This is a working paper of the Carnegie Mellon Project on Gender and Computer Science `https://www.cs.cmu.edu/afs/cs/project/gendergap/www/geekmyth.html`.

MARTIN, C. AND RAFALOW, M. 2015. Gendered Barriers to Participation in Gaming Culture. *In Proceedings of the Third Conference on GenderIT*, 49–52.

MARTINOVS, D. AND BARRETT, R. 2016. D2.4 — Interactive Teachers' Guide (Final version). *EU Delivery Agreement 645215*.

MARTINOVS, D., TINNEY, J., BOULTON, H., BELTRÁN, M. E., AND SPIELER, B. 2017. D3.2 — Game-making Teaching Framework (Final Version). *Grant Agreement number: 645215*.

MARTINSON, A. 2005. Playing with technology: Designing gender sensitive games to close the gender gap. *Working Paper SLISWP-03-05, School of Library and Information Science, Indiana University*.

MASON, C., KAHLE, J., AND GARDNER, A. 1991. Draw-A-Scientist Test: Future Implications. *In School Science and Mathematics 91*, 193–198.

MASTER, A., SAPNA, C., AND MELTZOFF, A. N. 2016. Computing Whether She Belongs: Stereotypes Undermine Girls' Interest and Sense of Belonging in Computer Science. *In Journal of Educational Psychology 108,* 3, 424–437.

MATLIN, M. 1999. Bimbos and Rambos: The Cognitve Basis of Gender Stereotypes. *In Eye on Psi Chi published by the national Honor Society in Psychology 3,* 2, 13–14.

MAYRING, P. 2014. Qualitative Content Analysis. Theoretical Foundation, Basic Procedures and Software Solution. *In basic procedures and software solution*. http://nbn-resolving.de/urn:nbn:de:0168-ssoar-395173.

MCGONIGAL, J. 2011. *Reality Is Broken: Why Games Make Us Better and How They Can Change the World*. In Penguin Press.

MCKNIGHT, P. E. AND NAJAB, J. 2010. Mann—Whitney U Test. *Irving B. Weiner W. Edward Craighead, Major Reference Works*.

MCLEAN, M. AND HARLOW, D. 2017. Designing Inclusive STEM Activities: A Comparison of Playful Interactive Experiences Across Gender. *In Proceedings of the 2017 Conference on Interaction Design and Children*, 567–574. New York, USA.

MCLEOD, S. 2008. Likert Scale. Retrieved February 18, 2018 from https://www.simplypsychology.org/likert-scale.html.

MCNULTY, K., ANTONELLI, M., BARTIK, J., HOLBER, F., MELTZER, M. W., SPENCE, F., AND TEITELBAUM, R. L. 1997. ENIAC Programmers. Retrieved March 20, 2018 from http://www.witi.com/center/witimuseum/halloffame/298369/ENIAC-Programmers-Kathleen-McNulty,-Mauchly-Antonelli,-Jean-Jennings-Bartik,-Frances-Synder-Holber-Marlyn-Wescoff-Meltzer,-Frances-Bilas-Spence-and-Ruth-Lichterman-Teitelbaum/.

MEDEL, P. AND POURNAGHSHBAND, V. 2017. Eliminating Gender Bias in Computer Science Education Materials. *In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, 411–416.

MEERBAUM-SALANT, O., ARMONI, M., AND BEN-ARI, M. 2010. Learning computer science concepts with scratch. *In Proceedings of the Sixth international workshop on Computing education research*, 69–76.

MELTON, M. 2017. A Majority of US Female Mobile Gamers Play Daily. And Candy Crush ranks as a favorite. Retrieved February 08, 2018 from https://www.emarketer.com/Article/Majority-of-US-Female-Mobile-Gamers-Play-Daily/1016458. eMarketer.

MERRIT, T. 2017. Top 5: Most in-demand IT jobs in 2017. Retrieved Janury 3, 2018 https://www.techrepublic.com/article/top-5-most-in-demand-it-jobs-in-2017/.

MICHAEL, D. AND CHEN, S. L. 2006. Serious Games — Games that educate, train, and inform. *In Course Technology Cengage Learning*.

MICROSOFT. 2017. Microsoft-Studie: Kreativität ist Schlüssel für mehr Frauen in MINT-Berufen. Retrieved February 08, 2018 from https://news.microsoft.com/de-de/microsoft-studie-mehr-frauen-mint-berufen/.

MILBERG, J. AND FUCHS, W. 2009. Nachwuchsbarometer Technikwissenschaften. *In Ergebnisbericht - Bundesministerium für Bildung und Forschung*.

MILGRAM, D. 2011. How to Recruit Women and Girls to the Science, Technology, Engineering, and Math (STEM). *In International Technology and Engineering Educators Association 71,* 3, 4–11.

MOCCOZET, L., TARDY, C., OPPRECH, W., AND LÉONARD, M. 2013. Gamification-based assessment of group work. *In International Conference on Interactive Collaborative Learning (ICL)*, 171–179.

MONITISE GROUP LIMITED. 2014. The mobile money landscape: Market Statistics and expert views. Retrieved February 08, 2018 from http://info.monitise.com/rs/clairmail/images/monitise_market-statistics-expert-views-2014.pdf.

MYANIMELIST. 2015. The Loyal Cats of Sailor Moon Crystal. Retrieved February 08, 2018 from https://myanimelist.net/featured/617/The_Loyal_Cats_of_Sailor_Moon_Crystal.

NAH, F. F.-H., ZENG, Q., TELAPROLU, V. R., AYYAPPA, A. P., AND ESCHENBRENNER, B. 2014. Gamification of Education: A Review of Literature. *In HCI in Business: First International Conference, HCIB 2014, Held as Part of HCI International 2014*, 401–409.

NAKAMURA, J. AND CSIKSZENTMIHALYI, M. 2009. The concept of flow. *In Snyder, C. R., & Lopez, S. J. (Ed.). Oxford handbook of positive psychology. Oxford University Press,*, 89–105.

NATIONAL SCIENCE FOUNDATION. 2018. Broadening Participation. Retrieved March 20, 2018 from http://www.nsf.gov/od/broadeningparticipation/bp.jsp.

NCWIT. 2015. NCWIT Fact Sheet. Retrieved Feburary 18, 2018 https://www.ncwit.org/ncwit-fact-sheet.

NCWIT. 2017. NCWIT's Women in IT: By the Numbers presents the most compelling statistics on women's participation in IT on a single page. Retrieved January 18, 2018 www.ncwit.org/bythenumbers. National Center for Women and Information Technolgy.

NEO, M. AND NEO, T. 2009. Engaging students in multimedia-mediated Constructivist learning – Students' perceptions. *In Educational Technology & Society 2*, 254–266.

NEWZOO. 2017. Male and Female Gamers: How Their Similarities and Differences Shape the Games Market. Retrieved February 26, 2018 from https://newzoo.com/insights/articles/male-and-female-gamers-how-their-similarities-and-differences-shape-the-games-market/.

NEXT GENERATION RECRUITMENT. 2018. Why aren't there more women in tech? Retrieved March 20, 2018 from https://www.nextgeneration.ie/blog/why-arent-there-more-women-in-tech/.

NICHOLS, T. E. AND HOLMES, A. P. 2001. Nonparametric Permutation Tests For Functional Neuroimaging: A Primer with Examples. *In Human Brain Mapping 15,* 1, 1–25.

NOOR-UL-AMIN, S. 2013. An Effective use of ICT for Education and Learning by Drawing on Worldwide Knowledge, Research, and Experience. Retrieved Feburary 07, 2018 http://www.nyu.edu/classes/keefer/waoe/amins.pdf. Research Scholar Department Of Education, University Of Kashmir.

OAKMAN, H. 2016. The computing curriculum - two years on. Retrieved February 08, 2018 https://edtechnology.co.uk/Article/the-computing-curriculum-two-years-on. Education Technology.

OCHSNER, A. 2015. Lessons Learned With Girls, Games, and Design. *In Proceedings of the Third Conference on GenderIT (GenderIT '15)*, 24–31. New York, USA.

OCLC. 2017. Online Computer Library Center, Inc; Ada Lovelace. Retrieved March 20, 2018 from https://www.worldcat.org/identities/lccn-n78030997/.

O'CONNOR, J. 2017. Women pioneered computer programming. Then men took their industry over. Retrieved March 20, 2018 from https://timeline.com/women-pioneered-computer-programming-then-men-took-their-industry-over-c2959b822523.

OECD. 2004. Problem solving for tomorrow's world : first measures of cross-curricular competencies from PISA 2003. http://www.pisa.oecd.org/dataoecd/25/12/34009000.pdf.

OFFICE FOR OFFICIAL PUBLICATIONS OF THE EUROPEAN COMMUNITIES. 1997. Treaty of Amsterdam amending the Treaty on European Union, the Treaties establishing the European Communities and certain related acts. Retrieved March 20, 2018 from http://www.europarl.europa.eu/topics/treaty/pdf/amst-en.pdf. ISBN 92-828-1652-4.

OFFICE OF INNOVATION AND IMPROVEMENT (OII). 2016. Computer Science for All Fact Sheet. Retrieved February 18, 2018 https://innovation.ed.gov/what-we-do/stem/computer-science/computer-science-for-all-fact-sheet/.

O'KELLY, J. AND GIBSON, J. 2006. Robocode & problem-based learning: a non-prescriptive approach to teaching programming. *In Proceedings SIGCSE Innovation and technology in computer science education*, 217–221.

OLGIATI, E. AND SHAPIRO, G. 2002. Promoting gender equality in the workplace. *Luxembourg: Office for Official Publications of the European Communities, 2002*. [online] http://www.eurofound.europa.eu/pubdocs/2001/61/en/1/ef0161en.pdf.

OLIVER, R. L. 1977. Effect of Expectation and Disconfirmation on Postexposure Product Evaluations — an Alternative Interpretation. *In Journal of Applied Psychology 62,* 4, 480.

OLIVER, T. 2017. An In-Depth Look At The Gender Gap in the Tech Industry. Retrieved February 04, 2018 from https://www.technicallycompatible.com/an-in-depth-look-at-the-gender-gap-in-the-tech-industry/.

ONG, P. 2016. Tides are turning in favour of women in tech. Here's why. Retrieved March 20, 2018 from https://www.techinasia.com/talk/programming-brighter-future-women-tech. Tech in Asia.

ONLINE COURSE REPORT. 2016. State of the MOOC 2016: A Year of Massive Landscape Change For Massive Open Online Courses. Retrieved Janury 15, 2018 https://www.onlinecoursereport.com/state-of-the-mooc-2016-a-year-of-massive-landscape-change-for--open-online-courses/.

ONWUEGBUZIE, A. J., DICKINSON, W. B., LEECH, N. L., AND ZORAN, A. G. 2009. A Qualitative Framework for Collecting and Analyzing Data in Focus Group Research. *In International Journal of Qualitative Methods 8,* 3, 1–21.

OR-BACH, R. AND LAVY., I. 2004. Cognitive activities of abstraction in object orientation: an empirical study. *In SIGCSE Bull 36,* 2, 82–86.

OWSTON, R., WIDEMAN, H., RONDA, N. S., AND BROWN, C. 2009. Computer game development as a literacy activity. *In Computer Education 53,* 3, 977–989.

PAASSEN, B., MORGENROTH, T., AND STRATEMEYER, M. 2017. What is a True Gamer? The Male Gamer Stereotype and the Marginalization of Women in Video Game Culture Sex Roles . *FEMINIST FORUM REVIEW ARTICLE 76*, 421–435.

PADEREWSKI, P., ARENAS, M. G., IRANZO, R. G., GONZÁLEZ, C., ORTIGOSA, E. M., AND PADILLA-ZEA, N. 2015. Bringing closer women to engineering: projects and strategies that promote their inclusion. *In Proceedings of the XVI International Conference on Human Computer Interaction* 37.

PAJARES, F. 2005. Gender differences in mathematics self-efficacy beliefs. *In A.M. Gallagher & J.C. Kaufman (Eds.), Gender differences in mathematics: An integrative psychological approach, Cambridge, United Kingdom: Cambridge University Press*, 294–315.

PAPERT, S. 1971. Teaching Children Thinking. *MIT AI Memo 247*.

PAPERT, S. 1985. *Mindstorms. Children, Computer, and Powerful Ideas*. In Basic Books Inc.

PAPERT, S. 1993. *The Children's Machine: Rethinking School In The Age Of The Computer: Bringing the Computer Revolution to Our Schools*. In BasicBooks.

PAPERT, S. 1998. Does Easy Do It? Children, Games, and Learning. *In Game Developer*.

PAPERT, S. AND HAREL, I. 1991. Constructionism. *New Jersey: Ablex Publishing Corporation*.

PARMAXI, A. AND ZAPHIRIS, P. 2014. Affordances of Social Technologies as Social Microworlds. In *In CHI '14 Extended Abstracts on Human Factors in Computing Systems*. http://dl.acm.org/citation.cfm?doid=2559206.2581267.

PASTERNAK, E., FENICHEL, R., AND MARSHALL, A. 2017. Tips for creating a block language with blockly. *In IEEE Blocks and Beyond Workshop (B&B), Raleigh*, 21–24.

PATRICK, H., RYAN, A., AND KAPLAN, A. 2007. Early Adolescents' Perceptions of the Classroom Social Environment. *In Journal of Educational Psychology 99,* 1, 83–98.

PAVLOV, I. 1927. Conditioned reflexes (G. V. Anrep,Trans.). *In London: Oxford University Press*.

PENN, G. 2005. Mightier than the soword; Developed Down Under - Why the Videogaming world wants a piece of Australia. Retrieved Feburary 6, 2018 `https://archive.org/stream/Edge-AUS-04-2005-01#page/n0/mode/2up`.

PEREIRA, O. AND RODRIGUES, J. 2013. Survey and analysis of current mobile learning applications and technologies. *In ACM Comput. Surv 46,* 2, 35.

PERRY, W. G. 1999. Forms of Ethical and Intellectual Development in the College Years. *San Francisco: Jossey-Bass Publishers.*.

PETRI, A., SCHINDLER, C., SLANY, W., AND SPIELER, B. 2015. Pocket Code Game Jams: a Constructionist Approach at Schools. *In Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*, 1207–1211.

PETRI, A., SCHINDLER, C., SLANY, W., AND SPIELER, B. 2016. Game Design with Pocket Code: Providing a Constructionist Environment for Girls in the School Context. *In Constructionism*.

PETRI, A., SCHINDLER, C., SLANY, W., SPIELER, B., BELTRÁN, M., URSA, Y., CABRERA-UMPIERREZ, M., ARREDONDO, M., AND DE LOS RIOS, S. 2015. Inclusive gaming creation by design in formal learning environments: 'girly-girls' user group in No One Left Behind. *In Design, User Experience, and Usability: Users and Interactions*, 153–161.

PETROVIÚC, P. 2016. On controlling LEGO Education platforms from Imagine Logo. *In Proceedings of Constructionism 2016*, 109–116. Bangkok, Thailand.

PEYTON-JONES, S. 2013. Computing in the national curriculum. A guide for primary teachers. *In Computing at School*.

PIAGET, J. 1968. Six Psychological Studie. *Anita Tenzer (Trans.), New York: Vintage Books*.

PIAGET, J. AND INHELDER, B. 1967. A Child's Conception of Space (F. J. Langdon & J. L. Lunzer, Trans.). *New York: Norton (Original work published 1948)*, 375–418.

POITSCHKE, T., ABLASSMEIER, M., RIGOLL, G., BARDINS, S., KOHLBECHER, S., AND SCHNEIDER, E. 2008. Contact-analog Information Representation in an Automotive Head-Up Display. *In Proceedings of the 2008 symposium on Eye tracking research & applications*, 119–122.

POLLOCK, L., MCCOY, K., CARBERRY, S., HUNDIGOPAL, N., AND YOU, X. 2004. Increasing high school girls' self confidence and awareness of CS through a positive summer experience. *In Proceedings of the 35th SIGCSE technical symposium on Computer science education (SIGCSE '04)*, 185–189.

POPULATIONPYRAMID.NET. 2018. Population Pyramids of the World from 1950 to 2100. Retrieved October 18, 2017 `https://www.populationpyramid.net/`.

PORTNOW, J. 2009. Opinion: Redefining Casual For The Hardcore. Retrieved Feburary 9, 2018 `https://www.gamasutra.com/php-bin/news_index.php?story=23249`.

PRESTON, J., CHASTINE, J., O'DONNELL, C., SAVANNAH, D., AND MACINTYRE, B. 2012. Game Jams: Community, Motivations, and Learning among Jammers. *In International Journal of Game-Based Learning 2,* 3, 51–70.

PROCTOR, C. AND BLIKSTEIN, P. 2016. Grounding How We Teach Programming in Why We Teach Programming. *In Proceedings of Constructionism 2016*, 109–116.

RAMNARINE-RIEKS, A. 2012. Learning through Game Design: An Investigation on the Effects in Library Instruction Sessions. *In Proceedings of the 2012 iConference*.

RAMOS, A. M. G. AND ROJAS-RAJS, T. 2016. Inclusion of Gender Perspective in Design and IT Environments. *In Proceedings of the XVII International Conference on Human Computer Interaction (Interaccion '16) 47*.

RASPBERRY PI FOUNDATION. 2017. About Code Club. Retrieved February 18, 2018 from `https://www.codeclub.org.uk/about`.

RAZAK, A. A., CONNOLLY, T., AND HAINEY, T. 2011. The use of Games-Based Learning Within the Curriculum for Excellence: The Teachers' Perspective. In In Proceedings of the European Conference on Games Based Learning. *Proceedings of the European Conference on Games Based Learning*. ISBN= 9781908272188.

REBOLLEDO-MENDEZ, G., AVRAMIDES, K., DE FREITAS, S., AND K., K. M. 2009. Societal impact of a serious game on raising public awareness: the case of FloodSim. *In Proceedings of the 2009 ACM SIGGRAPH Symposium on Video Games*, 15–22.

REPENNING, A., WEBB, D., KOH, K., NICKERSON, H., S.B. MILLER, C. B., AND HORSES, I. 2015. Scalable Game Design: A Strategy to Bring Systemic Computer Science Education to Schools through Game Design and Simulation Creation. *In Trans. Comput. Educ 15,* 2, 1–31.

RESNICK, M. 2017. Lifelong Kindergarten: Cultivating Creativity through Projects, Passion, Peers, and Play. *MIT Press*.

RESNICK, M., MALONEY, J., MONROY-HERNÁNDEZ, A., RUSK, N., EASTMOND, E., BRENNAN, K., MILLNER, A., ROSENBAUM, E., SILVER, J., SILVERMAN, B., AND KAFAI., Y. 2009. Scratch: programming for all. *In Commun. ACM 52,* 11, 60–67.

RICHARD, G. T. AND KAFAI., Y. B. 2016. Blind Spots in Youth DIY Programming: Examining Diversity in Creators, Content, and Comments within the Scratch Online Community. *In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 1473–1485.

RIPLEY, A. 2017. Boys Are Not Defective. Retrieved March 20, 2018 from `https://www.theatlantic.com/education/archive/2017/09/boys-are-not-defective/540204/`. The Atlantic Daily?

ROBINS, A., ROUNTREE, J., AND ROUNTREE, N. 2003. Learning and teaching programming: A review and discussion. *In Computer Science Education 13,* 2, 137–172.

ROELL, K. 2017. What is Grading on a Curve? Retrieved February 08, 2018 from `https://www.thoughtco.com/grading-on-a-curve-3212063`.

ROLLINGS, A. AND ADAMS, E. 2003. On Game Design. *In Indianapolis: New Riders*.

ROMEIKE, R. 2010. Didaktik der Informatik: Kriterien kreativen Informatikunterrichts. *Universität Potsdam*.

ROMERO, M. 2012. Learning through playing for or against each other? . *In Promoting collaborative learning in digital game based learning 5*, 15.

ROMISZOWSKI, A. J. 2004. How's the E-Learning Baby? Factors Leading to Success or Failure of an Educational Technology Innovation. *In Educational Technology 44,* 1, 5–27.

ROMMES, E., BOS, M., AND GEERDINK, J. 2011. Design and Use of Gender Specific and Stereotypical Toys. *International Journal of Gender, Science and Technology 3,* 1, 184–204.

ROSCHELLE, J. AND PEA, R. 2002. A walk on the wild side: how wireless handhelds may change CSCL. *Lawrence Erlbaum Associates, Mahwah*, 51–60.

ROUSE, R. 2001. Game Design: Theory and Practice. *Plano, Texas: Wordware*.

RTE. 2016. Girls start losing interest in science, maths at age 15. Retrieved October 18, 2017 `https://www.rte.ie/news/2016/1202/836019-girls-science-survey/`.

RYAN, R. AND DECI, E. 2000. Intrinsic and extrinsic motivations: Classic definitions and new directions. *In Contemporary Educational Psychology 25,* 1, 54–67.

SADLER, P. M., SONNERT, G., HAZARI, Z., AND TAI, R. 2012. Stability and volatility of STEM career interest in high school: A gender study. *In Science Education 96,* 3, 411–427.

SALEN, K. AND ZIMMERMAN, E. 2003. Rules of Play - Game Design Fundamentals. *In The MIT Press Cambridge Massachusetts*.

SAMPATH, D. 2004. Adaptive Object Re-CONfiguration: an approach to enhance, repeat playability of games and repeat watchability of movies. *In Proceedings of the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology*, 313–316.

SANCHEZ, G. 2013. PLS Path Modeling with R. *Emerald Group Publishing Limited*.

SANDERS, J. 2005. Gender and technology in education: what the research tells us. *In Proceedings of the international symposium on Women and ICT: creating global transformation* 6.

SANGRÀ, A. AND GONZRÀLEZ-SANMAMED, M. 2010. The role of information and communication technologies in improving teaching and learning processes in primary and secondary schools. *In Research in Learning Technology 18,* 3, 968–7769.

SCAMBOR, C. AND SCAMBOR, E. 2012. Intersektionale Analyse in der Praxis. Grundlagen und Vorgangsweise bei der Analyse quantitativer Daten aus der Intersectional Map. *In E. Scambor & F. Zimmer (Hrsg.), Die intersektionalle Stadt. Geschlechterforschugn und Medienkusnt an der Achsen der Ungleicheheit*, 43–78.

SCHELL, J. 2008. The Art of Game Design:A book of lenses. *In Morgan Kaufmann Pub*.

SCHIPFER, R. K. 2005. Der Wandel der Bevölkerungsstruktur in Österreich Auswirkungen auf Regionen und Kommunen. *Issue 51 of Papers, Österreichisches Institut für Familienforschung, ÖIF - Österr. Inst. für Familienforschung, 2005*.

SCHMUTZHART, I. 2012. gendup — Zentrum für Gender Studies und Frauenförderung. *Leitfaden f"ur einen gender gerechten Sprachgebrauch. Universität Salzburg*.

SCHNURR, B., BRUNNER-SPERDIN, A., AND STOKBURGER-SAUER, N. 2017. The effect of context attractiveness on product attractiveness and product quality: the moderating role of product familiarity. *In Marketing Letters 28,* 2, 241–253.

SCHÖN, S., EBNER, M., AND KUMAR, S. 2014. Implications of new digital gadgets, fabrication tools and spaces for creative learning and teaching. *Special edition 86*.

SCHUNK, D. H. 2014. Learning Theories, An Educational Perspective. *In Pearson Education Limited, Sixth Edition*.

SCHWARTZ, K. 2013. Giving Good Praise to Girls: What Messages Stick. Retrieved March 20, 2018 from `https://ww2.kqed.org/mindshift/2013/04/24/giving-good-praise-to-girls-what-messages-stick/`. KQED Inc.

SCHWARZER, R. AND JERUSALEM, M. 1995. Generalized Self-Efficacy scale. *In J. Weinman, S. Wright, & M. Johnston, Measures in health psychology: A user's portfolio. Causal and control beliefs*, 35–37. Windsor, UK: NFER-NELSON.

SCZESNY, S., FORMANOWICZ, M., AND MOSER, F. 2016. Can Gender-Fair Language Reduce Gender Stereotyping and Discrimination? *In Front Psychology 7,* 25.

SEABORN, K. AND DEBORAH, F. I. 2015. Gamification in theory and action. *In International Journal of Human-Computer Studies 74*, 14–31.

SEABORN, K. AND FELS, D. 2015. Gamification in theory and action: A survey. *In International Journal of Human-Computer Studie 74*, 14–31.

SELBY, C. 2012. Promoting computational thinking with programming. *In Proceedings of the 7th Workshop in Primary and Secondary Computing Education (WiPSCE '12)*, 74–77.

SENTENCE, S. AND CSIZMADI, A. 2015. Teachers' perspectives on successful strategies for teaching Computing in school. *In IFIP TC3 Working Conference*, 1–10.

SHAER, O., WESTENDORF, L., KNOUF, N. A., AND PEDERSON, C. 2017. Understanding Gaming Perceptions and Experiences in a Women's College Community. *In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 1544–1557.

SHALAWAY, L. 1998. Learning to Teach: – Not Just for Beginners: the Essential Guide for All Teachers. *Scholastic Inc.*.

SHETH, S. K., BELL, J. S., AND KAISER, G. E. 2012. Increasing Student Engagement in Software Engineering with Gamification. *Columbia University Computer Science Technical Reports*.

SHI, Y.-R. AND SHIH, J.-L. 2015. Game factors and game-based learning design model. *In International Journal of Computer Games Technology 11*, 1.

SHILLABEER, A. AND JACKSON, K. 2013. Gender imbalance in undergraduate IT programs — A Vietnamese perspective. *In Innovation in Teaching and Learning in Information and Computer Sciences 12,* 1, 70–83.

SHIRATUDDIN, N. AND ZAIBON, S. B. 2010. Mobile Game Based Learning with Local Content and Appealing Characters. *In International Journal of Mobile Learning and Organisation 4,* 1, 55–82.

SINGER, L. AND SCHNEIDER, K. 2012. It was a bit of a race: Gamification of version control. *In International Workshop on Games and Software Engineering 2*, 5–8.

SKINNER, B. 1976. About Behaviorism. *New York: Vintage Books*.

SLANY, W. 2014. Tinkering with Pocket Code, a Scratch-like programming app for your smartphone. *In Proceedings of Constructionism 2014*.

SMITH, J., MARTINOVS, D., BARRETT, R., BROWN, D., BOULTON, H., BURTON, A., AND BELTRÁN, M. E. 2016. D3.1 — Game-making Teaching Framework (Preliminary version). *EU Delivery Agreement 645215*.

SMITH, P. AND BOWERS, C. 2016. Improving Social Skills through Game Jam Participation. *In Proceedings of the International Conference on Game Jams, Hackathons, and Game Creation Events*, 8–14.

SPENCER, S. 2011. Universal Design for Learning: Assistance for Teachers in Today's Inclusive Classrooms. *In Interdisciplinary Journal of Teaching and Learning 1,* 1, 10–22.

SPENCER, S., STEELE, C., AND QUINN, D. 1999. Stereotype threat and women's math performance. *In Journal of Experimental Social Psychology 35,* 1, 4–28.

SPIEGEL. 2010. Studie: Ab 2015 fehlen Deutschland Arbeitskräfte. Retrieved December 09, 2017 `http://www.spiegel.de/wirtschaft/unternehmen/studie-ab-2015-fehlen-deutschland-arbeitskraefte-a-684805.html`.

SPIELER, B. 2018. Reinforcing Gender Equality by Analysing Female Teenagers' Performances in Coding Activities: A Lesson Learned. *In Proceedings of Conference on Gender IT 2018 / GEWINN-Konferenz 2018*.

SPIELER, B. AND MASHKINA, E. 2017. D5.4 — Report and findings from experimental pilot in Austria. *EU Delivery Agreement 645215*.

SPIELER, B., PETRI, A., BELTRAN, M. E., MÜNSTER, P., GAETA, E., AND REDONDO, T. 2017. D4.2 — Pocket Code functional specification framework and integration of transferred technologies (Final Version). *EU Delivery Agreement 645215*.

SPIELER, B., PETRI, A., SLANY, W., SCHINDLER, C., BELTRÁN, M. E., AND BOULTON, H. 2016. Pocket Code: A Mobile App for Game Jams to facilitate Classroom Learning through Game Creation. *In Proceedings of the Irish Conference on Game-Based Learning*, 61–79.

SPIELER, B., SCHINDLER, C., SLANY, W., AND MASHKINA, O. 2017. App Creation in Schools for different Curricula Subjects - Lessons Learned. *In Proceedings of the 9th International Conference on Education and New Learning Technologies*, 5814–5823. Barcelona, Spain.

SPIELER, B., SCHINDLER, C., SLANY, W., MASHKINA, O., BELTRÁN, M. E., BOULTON, H., AND BROWN, D. 2017. Evaluation of Game Templates to support Programming Activities in Schools. *In Proceedings of the 11th European Conference on Games Based Learning*, 600–609.

SPIVAK, G. 1990. The Post-Colonial Critique. Interviews, Strategies, Dialogues, Routledge. *London — New York*.

STANGL, W. 2018. Online-Enzyklopädie. Stichwort: 'Interesse'. Online Lexikon für Psychologie und Pädagogik. Retrieved March 03, 2018 from http://lexikon.stangl.eu/526/interesse/.

STARRUSS, I. 2010. Synopse zum Informatikunterricht in Deutschland. Analyse der informatischen Bildung an allgemein bildenden Schulen auf der Basis der im Jahr 2010 gültigen Lehrpläne und Richtlinien. Retrieved January 18, 2018 `https://dil.inf.tu-dresden.de/schulinformatik/informatikunterricht-in-deutschland/`.

STATISTA. 2016. Statistics of different entertainment markets. Retrieved February 9, 2018 `https://www.statista.com/topics/1680/gaming/` `https://www.statista.com/topics/1639/music/` `https://www.statista.com/topics/964/film/`.

STATISTA. 2017. Mobile Gaming Industry - Statistics & Facts. Retrieved February 08, 2018 `https://www.statista.com/topics/1906/mobile-gaming/`.

STATISTA MARKET ANALYTICS. 2016. Forecast of the smartphone user penetration rate in Austria, France, Germany, and United Kingdom (UK) from 2014 to 2021. Retrieved February 09, 2018 from `http://www.statista.com/statistics/567976/predicted-\discretionary{-}{}{}smartphone-user-penetration-rate-in-austria` `http://www.statista.com/statistics/568093/predicted-smartphone-user-penetration-rate\discretionary{-}{}{}in-france/` `http://www.statista.com/statistics/568095/predicted-smartphone-user-\discretionary{-}{}{}penetration-rate-in-germany/` `http://www.statista.com/statistics/553707/predicted-smartphone-user-\discretionary{-}{}{}penetration-rate-in-the-united-kingdom-uk/`.

STOET, G. AND GEARY, D. C. 2018. The Gender-Equality Paradox in Science, Technology, Engineering, and Mathematics Education. *In Psychological Science*.

STOUT, J. AND CAMP, T. 2014. Now what?: action items from social science research to bridge the gender gap in computing research. *In SIGCAS Comput. Soc. 44,* 4, 5–8.

STOUT, J. G., DASGUPTA, N., HUNSINGER, M., AND MCMANUS, M. A. 2011. STEMing the tide: Using ingroup experts to inoculate women's self-concept in science, technology, engineering, and mathematics (STEM). *In Pers. Soc. Psych 100*, 255–270.

STRUNGÚA, A. 2014. Osgood's semantic differential: a review of the Romanian social sciences literature. *In Social Sciences and Education Research Review 2 2*, 22–28. www.sserr.ro.

SUBSOL, G. 2005. Virtual Storytelling — using virtual reality technogies for storytelling. *In Proceedings of the third international conference*, 251–259. the rapunsel project, Springer Verlag Berlin Heideberg.

SUDDABY, P. 2013. 16 Tips, Tools and Resources for Your Next Game Jam . Retrieved December 23, 2017 http://gamedevelopment.tutsplus.com/articles/16-tips-tools-and-resources-for-your-next-game-jam--gamedev-12084.

SUKUENSEONG, D. 2016. Usability Guidelines for Designing Mobile Learning portals. *In Mobility 06*, 25–27.

SULLIVAN, G. M. AND ARTINO, A. R. 2013. Analyzing and Interpreting Data From Likert-Type Scales. *In Journal of Gradate Medical Education 5,* 4, 541–542.

SUTHERLAND, J. AND SCHWABER, K. 1995. Business object design and implementation. *In OOPSLA Workshop Proceedings. The University of Michigan*.

SWARTOUT, W. AND VAN LENT, M. 2005. Making a game of systemdesign. *In Communications of the ACM 46,* 7, 32–39.

SYAMSUL, Z. AND NORSHUHADA, S. 2010. Adapting Learning Theories in Mobile Game-Based Learning. *In Digital Game and Intelligent Toy Enhanced Learning (DIGITEL), 2010 Third IEEE International Conference*, 124–128.

SZEKERES, E. K. 2005. Sprachlicher Sexismus und sprachliches Gender Mainstreaming im Fokus der EuropÃd'ischen Sprachpolitik. *In European Integration Studies 4,* 2, 25–44.

TAKAHASHI, D. 2017. Sensor Tower: Mobile game revenues grew 32% in Q2 as Asian titles surged. Retrieved February 06, 2018 https://venturebeat.com/2017/07/24/mobile-game-revenues-grew-32-in-q2-as-asian-titles-surged/.

TANDON, N. 2012. A bright future in ICT opportunities for a new generation of women. *In digital inclusion. A Bright Future in ICTs OPPORTUNITIES FOR A New Generation of Women*. https://www.itu.int/en/ITU-D/Digital-Inclusion/Women-and-Girls/Documents/ReportsModules/ITUBrightFutureforWomeninICT-English.pdf.

TARKUS, A., GLATZ, D., MAUTHNER, K., PETRI, A., AND SLANY, W. 2016. An Approach to evaluate Technolog Acceptance based on the Example of the Educational App Pocket Code. *In Proceedings of the 18th GENERAL ONLINE RESEARCH CONFERENCE 72*.

TECHOPEDIA. 2018. Arcade Game. Retrieved February 9, 2018 https://www.techopedia.com/definition/1903/arcade-game.

TEDRE, M. AND DENNING, P. 2016. The Long Quest for Computational Thinking . *In Proceedings of the 16th Koli Calling Conference*, 120–129.

TELSTRA. 2015. Mobile Protect Fact Sheet. http://exchange.telstra.com.au/wp-content/uploads/2015/03/Mobile-Protect-Fact-Sheet-FINAL1.pdf. Last visited: April 21st 2015.

THE ASSOCIATION FOR WOMEN IN SCIENCE/AWIS. 2017. Inspiring Stories from Real STEM Heroes. Making the world we share a better place. Retrieved February 20, 2018 from https://www.awis.org/member-spotlights/. Washington, DC.

THE CENTER FOR EMOTIONAL HEALTH. 2016. Behavioral Assessments. Retrieved January 18, 2018 from http://www.thecenterforemotionalhealth.com/behavioral-assessments.

THE DEMING INSTITUTE. 2018. PDSA Cycle. Retrieved February 20, 2018 from https://deming.org/explore/p-d-s-a.

THE NATIONAL CENTER FOR WOMEN & INFORMATION TECHNOLOGY (NCWIT). 2016. Women in IT: The Facts Infographic. Retrieved March 20, 2018 from https://www.ncwit.org/resources/women-it-facts-infographic-2016-update.

THE WHITE HOUSE. 2016. Obama's Computer Science for All Initiative Is Gaining Momentum. Retrieved Janury 8, 2018 from https://cacm.acm.org/careers/207286-obamas-computer-science-for-all-initiative-is-gaining-momentum/fulltext.

THE WOMEN'S ENGINEERING SOCIETY (WES). 2017. Useful Statistics. Retrieved October 10, 2017 http://www.wes.org.uk/statistics.

THE WORLD BANK. 2017. Labor force, female (procent of total labor force). Retrieved February 18, 2018 from https://data.worldbank.org/.

TINKER, R. AND PAPERT, S. 1988. Tools for science education. In J. D. Ellis, editor, Information Technology and Science Education. *1988 AETS Yearbook* 1, 1–23. Columbus: SMEAC Information Reference Center (SMEAC/IRC), The Ohio State University, 1989.

TINNEY, J., BROWN, D. J., AND BURTON, A. 2015. D.1.1 — Characterization of the Education Domain . *EU Delivery Agreement 645215*.

TOMB RIDER. 2017. Evalution of Tomb Rider Lara Croft. Retrieved February 04, 2018 from https://femhype.files.wordpress.com/2015/05/evolution.png.

TOWNSEND, G. C. 2002. People who make a difference: mentors and role models. *In SIGCSE Bull 34,* 2, 57–61.

TRAUTH, E. M., QUESENBERRY, J. L., AND MORGAN, A. J. 2004. Understanding the under representation of women in IT: toward a theory of individual differences. *In Proceedings of the 2004 SIGMIS conference on Computer personnel research: Careers, culture, and ethics in a networked environment*, 114–119. New York, USA.

TSAN, J., BOYER, K. E., AND LYNCH, C. F. 2016. How Early Does the CS Gender Gap Emerge?: A Study of Collaborative project Solving in 5th Grade Computer Science. *In Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, 388–393. NY, USA.

TUMLIN, N. 2017. Teacher Configurable Coding Challenges for Block Languages. *In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, 783–784.

UNESCO. 2015. STEM and Gender Advancement. Retrieved February 04, 2018 from http://uis.unesco.org/en/news/stem-and-gender-advancement.

UNESCO ASIA-PACIFIC. 2017. Closing the gender gap in STEM. *United Nations*, 1–4.

UNFRIED, A., DA FABER, M., STANHOPE, D. S., AND WIEBE, E. 2015. The Development and Validation of a Measure of Student Attitudes Toward Science, Technology, Engineering, and Math (S-STEM). *In Journal of Psychoeducational Assessment 33,* 7, 622–639.

UNITED NATIONS. 2016. Leaving no one behind: the imperative of inclusive development. *In Report on the World Social Situation 2016. Department of Economic and Social Affairs.*

USAID. 2008. Education form a gender equality perspective. *USAID's Office of Women in Development Achieving Equality in Education.*.

USER INTERFACE DESIGN GMBH. 2018. AttrakDiff. Retrieved January 03, 2018 from http://attrakdiff.de/.

UX PROFESSIONALS' ASSOCIATION. 2005-2012. The Usability Body of Knowledge. Retrieved February 08, 2018 from http://www.usabilitybok.org/glossary/19. Last access 13.04.2016.

VEILLEUX, N., BATES, R., ALLENDOERFER, C., JONES, D., CRAWFORD, J., AND SMITH, T. F. 2013. The relationship between belonging and ability in computer science. *In Proceeding of the 44th ACM technical symposium on Computer science education*, 65–70.

VERTO ANALYTICS. 2015. Who Plays Mobile Games and When? Retrieved February 08, 2018 from http://www.vertoanalytics.com/who-plays-mobile-games-and-when/.

VERVECKEN, D. AND HANNOVER, B. 2012. Ambassadors of gender equality? How use of pair forms versus masculines as generics impacts percept ion of the speaker. *In European Journal of Social Psychology 42*, 754–762.

VERVECKEN, D. AND HANNOVER, B. 2015. Yes I can! The impact of gender fair descriptions of traditionally male occupations on children's perceptions of job status, job difficulty and vocational self-efficacy beliefs. *In Social Psychology 46*, 76–92.

VERVECKEN, D., HANNOVER, B., AND I., I. W. 2013. Changing (s)expectations: how gender fair job descriptions impact children'vs perceptions and interest regarding traditionally male occupations. *In J. Vocat. Behav 82*, 208–220.

VON FÖRSTER, H., VON GLASERSFELD, E., HEIJL, P. M., J.SCHMIDT, S., AND WATZLAWICK, P. 2009. Einführung in den Konstruktivismus. *Heinz Gumin und Heinrich Meier, 11.Auflage, München*.

VON GLASERSFELD, E. 1995. A constructivist approach to teaching. *Steffe L.P. & Gale J. (eds.) Constructivism in education. Erlbaum, Hillsdale: 3-15. Available at http://www.vonglasersfeld.com/172*.

VUORIKARI, R., PUNIE, Y., CARRETERO, S., AND DEN BRANDE, L. V. 2016. DigComp 2.0: The Digital Competence Framework for Citizens. Retrieved Feburary 05, 2018 https://tinyurl.com/zmjb3fb.

VYGOTSKY, L. 1978. Mind in Society. *London: Harvard University Press*.

WALBY, S. 2005. Gender Mainstreaming: Productive Tensions in Theory and Practice Sylvia Walby Social Politics: International Studies in Gender. *In Social Politics: International Studies in Gender, State & Society 12,* 3, 321–343.

WALTON, G. M. AND COHEN, G. L. 2007. A Question of Belonging: Race, Social Fit, and Achievement. *In Journal of Personality and Social Psychology 92,* 1, 82–96.

WARDELL, S. 2016. *An Investigation of Creative Environments in the Western World*. In Lambent Interfaces.

WATSON, J. AND RAYNER, R. 1920. Conditioned emotional reactions. *In Journal of Experimental Psychology 3*, 1–14.

WEARN, N. AND MCDONALD, B. 2016. Ethos of location and its implication to the motivators of Global Games Jam participants. *In Proceedings of the International Conference on Game Jams, Hackathons, and Game Creation Events*, 58–61.

WEIBERT, A., VON REKOWSKI, T., AND FESTL, L. 2012. Accessing IT: a curricular approach for girls. *In Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design*, 785–786.

WEINTROP, D. AND WILENSKY, U. 2015. To block or not to block, that is the question: students' perceptions of blocks-based programming. *In Proceedings of the 14th International Conference on Interaction Design and Children*, 199–208.

WICS AVOCACY COUNCIL. 2015. Gender in Tech at Harvard. Retrieved March 20, 2018 from `http://advocacy.harvardwics.com./data/`. In Harvard Women in Computer Science.

WILLIAMS, D., MARTINS, N., CONSALVO, M., AND IVORY, J. D. 2009. The virtual census: representations of gender, race and age in video games. *In New Media & Society 11,* 5, 815–834.

WILLIAMS, G. 2014. Are you sure your software is gender-neutral? *In Magazine interactions 21,* 1, 36–39.

WING, J. 2006. Computational thinking. *In Communications of the ACM 49,* 3, 33–35.

WING, J. 2008. Computational thinking and thinking about computing. *In Philosophical Transactions of the RoyalSociety 36,* 1881, 3717–3725.

WISNIEWSKI, T. 2017. THE FUTURE TECH WORKFORCE: BREAKING GENDER BARRIERS. Retrieved February 18, 2018 from `http://www.isaca.org/SiteCollectionDocuments/Breaking-Gender-Barriers_res_eng_0317.PDF`. ISACA.

WOLBER, D. 2009. App inventor and real-world motivation. *In Proceedings of the 42nd ACM technical symposium on Computer science education*, 601–606.

WONG, B. AND KEMP, P. 2017. Technical boys and creative girls: the career aspirations of digitally—-skilledyouths. *In The Cambridge Journal of Education*.

WOOD, S. AND MAYO-WILSON, E. 2012. School-based mentoring for adolescents: A systematic review and meta-analysis. *In Research on Social Work Practice 22,* 3, 257–269.

WOOLLEY, A. W., CHABRIS, C. F., PENTLAND, A., HASHMI, N., AND MALONE, T. W. 2010. Evidence for a collective intelligence factor in the performance of human groups. *In Science 330,* 6004, 686–688.

WORLD ECONOMIC FORUM. 2016. The Future of Jobs - Employment, Skills and Workforce Strategy for the Fourth Industrial Revolution. *In Global Challenge Insight Report*. `http://www3.weforum.org/docs/WEF_Future_of_Jobs.pdf`.

WU, B. AND WANG, A. I. 2012. A Guideline for Game Development-Based Learning: A Literature Review. *In International Journal of Computer Games Technology 2012*.

XINOGALOS, S., SATRATZEMI, M., AND DAGDILELIS, V. 2006. An introduction to object-oriented programming with a didactic microworld. *In Computers & Education 47,* 2, 148–171.

YEE, N. 2017a. Beyond 50/50: Breaking Down The Percentage of Female Gamers by Genre. Retrieved February 26, 2018 from `https://quanticfoundry.com/2017/01/19/female-gamers-by-genre/`. Quantic Foundry.

YEE, N. 2017b. Just How Important Are Female Protagonists? Retrieved October 18, 2017 https://quanticfoundry.com/2017/08/29/just-important-female-protagonists/.

YELLAND, N. 1995. Collaboration and learning with Logo: does gender make a difference? *In The first international Conference on Computer support for collaborative learning*, 397–401. NJ, USA.

YOUNG, D. M., RUDMAN, L. A., BUETTNER, H. M., AND MCLEAN, M. C. 2013. The Influence of Female Role Models on Women's Implicit Science Cognitions. *In Psychology of Women Quarterly 37,* 3, 283–292.

ZAGAMI, J., BODEN, M., KEANE, T., MORETON, B., AND SCHULZ, K. 2015. Girls and computing: Female participation in computing in schools. *In Australian Educational Computing 30,* 2, 1–14. http://journal.acce.edu.au/index.php/AEC/article/view/79.

ZAIDI, R., FREIHOFER, I., AND TOWNSEND, G. 2016. Using Scratch and Female Role Models while Storytelling Improves Fifth-Grade Students' Attitudes toward Computing. *In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE '17)*, 791–792.

ZICHERMANN, G. AND CUNNINGHAM, C. 2011. Gamification by design: Implementing game mechanics in web and mobile apps. *In O'Reilly Media, Inc.*

ZIESEMER, A. 2013. Gamification Aware: Users Perception About Game Elements on Non-Game Context. *In Brazilian Symposium on Human Factors in Computing Systems.*

ZIMMERMAN, D. W. 2014. Comparative Power of Student T Test and Mann-Whitney U Test for Unequal Sample Sizes and Variances. *In the Journal of Experimental Education 55,* 3, 171–174.

Appendix A

Appendix

A.1. Example Lesson Plan

Pocket Code Units – Fine Arts/German – Grade 7 – Akademisches Gymnasium

<table>
<tr><th colspan="6">Context & Profile</th></tr>
<tr><td>Author:
TU Graz</td><td>Title: Applying Pocket Code to Fine Arts & German: debugging a precoded template and including own vocabulary</td><td>Timescale:
8 units at 45 min</td><td colspan="2">Year group/age:
Grade 6</td><td>No in group:
29</td></tr>
<tr><td colspan="6">Relevant contextual information on learners:
17 different mother tongues (multicultural) among students</td></tr>
<tr><th colspan="2">How does this lesson fit into the subject curriculum or the wider curriculum?</th><th colspan="4">Prior learning of learners</th></tr>
<tr><td colspan="2">Arts:
Asset creation, handmade artefacts, visual media, multimedia about the topic “Alice in Wonderland”

German:
Memorizing, basic vocabulary training, spelling

Computing:
Understand fundamental principles of programming (loops, broadcast messages, variables), support of computational thinking

Math:
Use of coordinates, understanding the use of variables</td><td colspan="4">Students have read the Alice in Wonderland novel by Lewis Carroll in German class before the first Pocket Code unit. It is expected that students have created their own Alice characters before the first Pocket Code lessons in fine arts.
Students have been introduced to the basic functions of Pocket Code. They are already skilled in:
Pocket Paint usage
Importing graphical assets (Media Library, phone gallery)
Use of control bricks
Use of forever and condition loops
Sending and receiving broadcast messages
Animations and appearance (move/look bricks)
Playing and recording sounds
Use of variables</td></tr>
<tr><th colspan="6">The Learning</th></tr>
<tr><th>Groups</th><th>Intended progress (Learning Objectives)</th><th colspan="2">How will this progress be demonstrated?</th><th colspan="2">Assessment of progress by…</th></tr>
<tr><td>All</td><td>Learning goal: Enhance a game with contents learned in class, your own vocabulary, and assets by applying the puzzle template
Computing: Identifying how to the program was developed; debug the program and include your own code bricks to add assets and vocabulary to the game
German: Learn and memorize new vocabulary of fruits in different mother tongues spoken in class.</td><td colspan="2">By the end of the session, students created a personalized vocabulary game by debugging and enhancing the puzzle template. They will have understood parts of the original code, added the logic to the game play, and customized the graphics. The finished game will be presented to their peers and teacher.</td><td colspan="2">Games are completed, with significant personalization (including vocabulary of their own mother tongue, own assets).
Games can be played by peers.
Students can memorize new vocabulary in another language.
Cultural exchange among students.
Games given peer/teacher feedback.</td></tr>
<tr><th colspan="6">Organization</th></tr>
<tr><td colspan="2">Resources:
Screen casting with “All Cast Dongle” and projector
1 tablet per small group of students
1 template
Tutorial cards
Support:
Peer support, NOLB team support</td><td colspan="4">Working with others: Working in groups of six (3 groups of two) with one teacher assistant.
Learners:
Students 11 to 12 years old
No special needs
Sitting in teams of small circles</td></tr>
</table>

Unit 1 - 2 (45 min)			
Timings	**Content**		
To start with…		**Cognitive /Behavioral***	**Learning scenario***
10 min	Instructions and presentation of the template	B	SG
Main learning			

80 min	Use the precoded template, studying the code, and debug it.
	Personalize the game:
	• Adding their own Alice character
	• Adding their own fruit objects
	Add missing functionality:
	• Control of Alice character by using the inclination sensors
	  ○ learn about sensors
	  ○ troubleshooting: mirroring the sensors and set the speed of the movement for a smoother movement
	• Moving background: learn about coordinates and how to place objects
	• Fruits glide from one side to the other
	  ○ learn about gliding objects in a loop
	  ○ learn about using the random function
		B/C	SG

Unit 3- 4 (45 min)			
Timings	**Content**		
To start with…		**Cognitive /Behavioral***	**Learning scenario***
5 min	Teacher provides a brief summary of the previous unit. Students continue where they left off last time.	C	SG
Main learning			
85 min	Students continue with adding logic to their game: • integrate a collision detection between their character and the gliding objects • learn about how to make a collision by using variables and functions • learn about the difference between local/global variables	B/C	SG

Unit 5-8 (45 min)			
Timings	**Content**		
Main learning		**Cognitive /Behavioral***	**Learning scenario***
90 min	Finalize the games (adding parts of the "Shape of a Game"): • add a score: repetition of variables to define the score ○ learn how to display a score and change the variable • add a win screen and a game over screen	B/C	SG
Plenary			
90 min	Students will show the class the games developed during the session. The teacher will evaluate the games in collaboration with the class. Example game:	B	FG

	 Select a language  Instructions screen  List of variables  Game: catch the fruits		

*Learning Scenario key: FG – Full group
SG – Small group (including partners)
I - Individually

*Cognitive/Behavioral Key:
C – Cognitive
B – Behavioral

A.2. Informed Consents NOLB

Graz University of Technology

Institute for Software Technology
University of Technology Graz
Inffeldgasse 16b
A-8010 Graz
Austria / Europe

Univ.-Prof. Dipl.-Ing. Dr.techn.
Wolfgang Slany
Head of department

Phone: +43 - 316 - 873 - 5721
Fax: +43 - 316 - 873 - 5706

wolfgang.slany@tugraz.at
http://www.ist.tugraz.at/

Administration:
Petra Pichler
Tel: +43 - 316 - 873 - 5711
Elisabeth Orthofer
Tel: +43 - 316 - 873 - 5713
sekretariat@ist.tugraz.at

DVR: 008 1833 UID: ATU 574 77 929

Graz, 14/02/2017

Dear parents and legal guardians,

Your child is being asked to participate in a pilot study as part of the EU project "No One Left Behind", co-funded by H2020 programs. This project is conducted by the Technical University Graz and our partners in Spain (INMARK Europe, UPM), England (GameCity, Nottingham Trent University) and Germany (Hochschule der Medien). The objective/goal of this study is to integrate our mobile learning application Pocket Code and Create@School into the classroom. For further information visit our website: www.no1leftbehind.eu. Both apps have been initiated and developed at Graz University of Technology especially for children between 11 and 17 years. Pocket Code is freely available for Android at the Google Play Store (http://catrob.at/pc) in German, English, and in many other languages. Create@School is a version especially for schools and therefore will be first validated by our partner schools as an open beta version. Within this version special events and time durations will be tracked to analyze different parameters of the created games. Only the teachers themselves will have access to these tracked data. For the purpose of evaluation the anonymized data will be used.

The apps use a visual programming language, which allows to create own games, animations, interactive music videos, and many other app types directly on smartphones or tablets. Similar to Scratch, programs are created by snapping together command bricks (“lego-style”). It’s a tool for novices to learn program flows and basic logic concepts. Furthermore, the apps encourage students to work creatively and collaboratively in teams.

Create@School is now used for the 2nd cycle of the project. The goal of this cycle is to evaluate this version for school purposes and to test if it fulfils the requirements. On the one hand, the predefined templates will be tested and, on the other hand, new templates will be created with the help of the students. These templates should facilitate the students’ entry into the world of programming. The app will be used as a supporting learning tool during project work in schools from October till June 2017.

It is important for us to observe how students learn with this software and what should be improved. For this purpose we are planning to conduct interviews regularly to gather information and feedback. During the course of this study, we will conduct follow-up interviews, questionnaires and observations, take photos and make short videos. This information will be used for scientific analyses, papers and public purposes. The researcher(s) retain(s) the right to use and publish non-identifiable data. When the results of this research are published or discussed in conferences, no information will be included that would reveal your child's identity.

If you have questions or concerns during the time of your child's participation in this study, or after its completion, please contact Univ.-Prof. Wolfgang Slany.

Should you decide to allow your child to participate in this research study, you will be asked to sign this consent form once all your questions have been answered to your satisfaction.

Best regards,

Univ.-Prof. Dipl.-Ing. Dr. Wolfgang Slany

Informed consent:

I have read this consent form and I understand what is expected from my child as a participant in this study. I give consent for my child to be a part of the research study including follow-up interviews, questionnaires, observations, videos and photos.

Name of child: ________________________________ Year of birth (e.g. 1997): ________

Name of parent/guardian: __________________________________

Phone: _____________________ Email: ____

Date: ______________________ Signature: ______________________

We are very thankful that you allow your child to take part in our research study. To allow us to document your child's use of our software by taking pictures and videos, please tick the appropriate box:

☐ Yes, I allow the use of video and photo recordings of my child like described above.
☐ I allow to use videos and photos of my child if the face is unrecognized by "pixelating".
☐ No, you are not allowed to use pictures and videos in which my child appears.

A.3. Attractive Survey

Attractiveness of Create@School

Age:

Gender:

School:

Country:

With the help of the word-pairs, please enter what you consider the most appropriate description for Create@School. Please tick off one item in every line

human*	○	○	○	○	○	○	○	technical
isolating*	○	○	○	○	○	○	○	connective
pleasant*	○	○	○	○	○	○	○	unpleasant
inventive*	○	○	○	○	○	○	○	conventional
simple*	○	○	○	○	○	○	○	complicated
professional*	○	○	○	○	○	○	○	unprofessional
ugly*	○	○	○	○	○	○	○	attractive
practical*	○	○	○	○	○	○	○	impractical
likeable*	○	○	○	○	○	○	○	disagreeable
cumbersome*	○	○	○	○	○	○	○	straightforward
stylish*	○	○	○	○	○	○	○	tacky
predictable*	○	○	○	○	○	○	○	unpredictable
cheap*	○	○	○	○	○	○	○	premium
alienating*	○	○	○	○	○	○	○	integrating
brings me closer to people*	○	○	○	○	○	○	○	separates me from people
unpresentable*	○	○	○	○	○	○	○	presentable
rejecting*	○	○	○	○	○	○	○	inviting
unimaginative*	○	○	○	○	○	○	○	creative
good*	○	○	○	○	○	○	○	bad
confusing*	○	○	○	○	○	○	○	clearly structured
repelling*	○	○	○	○	○	○	○	appealing
bold*	○	○	○	○	○	○	○	cautious
innovative*	○	○	○	○	○	○	○	conservative
dull*	○	○	○	○	○	○	○	captivating
undemanding*	○	○	○	○	○	○	○	challenging
motivating*	○	○	○	○	○	○	○	discouraging
novel*	○	○	○	○	○	○	○	ordinary
unruly*	○	○	○	○	○	○	○	manageable

A.4. Guidelines for PECC-Activities

Guidelines for PECC-Activities

Items	Tasks	PECC Category	
	STAGE 0: Preparation		
(0.1) Structure of the coding course	**0.1.1: Choose your target group** • Gender, age (12-15 years old) • In or outside the classroom (after school program) • Youth center, coding camps, etc.	**Coding – Structure**	☐
	0.1.2: Determine the available units Recommendation: • Instruction (the Starter, 1-2 units) • Game design (The Main Learning, 2-4 units) • Coding (The Main Learning, 4-8 units) • Presentation (The Closing, 1-2 units)	**Creativity – Structure**	☐
	0.1.3: Choose suitable tools **For coding:** • Visual-based coding (e.g., Pocket Code, Scratch, Snap) or robotics (e.g., Lego Mindstorms) • Text-based coding (e.g., Text Editors, Eclipse, Android Studio, etc.) **For creating game assets (how they are produced)*** • Artwork (by hand) • Tools, e.g. Photoshop, InDesign • Assets from the internet (be aware of copyright issues!) • Personal photographs • Use assets from available media libraries • Sound design: personal records, internet		☐
	0.1.4: Define the learning goal(s)🕭 or a general goal for games A (learning) goal consists of three parts: action, content, and condition. **The (learning) goals need to be defined according to** 🕭 **(a)** Learning goals/objective of the curriculum subject in which coding is applied <u>Example</u>: "Add 5 questions about the `French Revolution' to your game". **(b)** Learning goals for game design/coding <u>Example</u>: "Integrate min. 2 objects designed by yourself (artwork)"	**Coding – Teaching Approach**	☐
	0.1.5 Choose the engagement level * Group constellations (homogeneous/heterogeneous teams) (1) small groups (2-5) (2) pair work (3) work individually (but all working on the same learning goal)		☐
(0.2) Prepare your material	**1.1.4 Create tailored challenges *** (a) Template/Framework: students start with a pre-coded game and to add code/assets to finalize it (also allows customization, etc.) (b) Learning-by-doing: provide tutorials, helpful material/prepared functions, guidance ✎	**Engagement – Collaboration**	☐
	1.1.5 Set-up & Prepare • Presentation, if needed • Print storyboards (→ see storyboard)	**Coding – Structure**	☐

	• Handicraft items for “brick on paper” activity (scissors, tape, paper, etc.) • Template/framework program, example games to present, etc. • Setup platforms/tools (accounts, installation, charge mobile devices) • Other (room, date, time, equipment e.g., projector, etc.)		
STAGE 1: Introduction			
(1.1) Create a realistic picture of STEM jobs	**1.1.1 Create a safe environment *** Allow/ask questions, spark discussions In small groups/with the whole class: • Which technical professions do you know? • Which study/training do you need to acquire technical skills? • What does a computer scientist do? Do you know people who are working in those fields? • Who already has experience in coding? Which tools did you use for coding? • What is coding? What is an algorithm? • Which programming languages do you know? **1.1.2 Visit companies, invite role models *** ✂ **Be a role model/mentor on your own!** • Asking for resources to promote the improvement of technological knowledge (companies, universities) • Establishing direct communication between STEM professionals and students • Invite STEM professionals (role models) from the industry/university • Tell about role models and famous women who have succeed in computer science (e.g., Ada Lovelance) • Inspire students for STEM • Address the issue: Why do you think there are fewer women in IT than men? **1.1.3 Understand the learner’s playing behavior *** • What kind of games do you play? • What makes you play games? • What hinders you from playing games?	**Engagement – Warm up** **Playing - Play**	☐ ☐ ☐
(1.2) Provide a convenient starting point	**1.2.1 Design Learning: What is/How do _________?** Students are not familiar with “coding vocabulary” and practices. The most important terms are (→ IT Glossary) • Loops, conditions, variables, data types, objects, pseudocode, conditionals, function, iteration, parameter, broadcast messages, etc. It is not necessary to explain all of them, ask them if they are familiar with these concepts. Explain why they are needed (e.g., for creating a score, you need to define a variable; in order for objects to interact, you need messages). **The answer is: Engagement!** (a) integrate important functionalities in the example program (next step), so students can see what they are needed for (b) prepare a presentation with showcases/example programs (c) do “Unplugged Coding”, e.g.	**Engagement – Collaboration** **Coding – Structure**	☐

	○ "Program" a classmate like a robot (start/end point) ○ Paint "instructions" ○ Pack a rucksack with "variables" ○ Send "broadcasts" through the classroom		
	1.2.1 Introduce the tools or let students explore * • Show the UI, menu, and structure of the tool/platform • Show them where to find help, tutorials, useful forums/groups, demos (e.g., on YouTube) Coding: Starter program: ✂ (1) Create a collaborative program with the whole class (e.g. on the projector) which covers important steps (about the program they are going to create on their own): e.g., add an object, movement, interaction, etc. (a) One or two students come to the front of the class and add one small but meaningful step to the game (class is allowed to help) (b) Ask students for the next step while programming (2) Let students program a game (a small starter task) with the help of tutorials (guides, step-to-step) and let them add enhancements, e.g., add an animation, add a sound, score, etc.	**Coding – Personal Experiences**	☐
	1.2.2 Don't forget the fun! - Let students PLAY * ✂ (a) Show students example games (b) Let them play games on their own (i.e. featured games, best practice)	**Playing - Play**	☐
STAGE 2: Story & Game Design			
(2.1) Foster self-directed learning to create personal experiences	**2.1.1 Bring "Freedom of Choice" to your course to create a sense of ownership *** • Designing of personal games from scratch: (a) Don't restrict the game design at all, let them choose game elements, e.g. story, genre, theme, goal, MDAs, assets (b) Define a frame, e.g., use of certain properties, genre, design elements, or MDAs • Use of templates: allow customization, personalization, and enhancements	**Playing** **Engagement** **Creativity** **Coding**	☐
	2.1.2 Let's get it started! * • Describe the activity: task, structure, units –> strive for mutual understanding • Explain the (learning) goal: define a sub-goal for each unit • Support the formation of homogeneous groups	**Coding – Coding**	☐
(2.2) Bring in the gaming/design elements	**2.2.1 Give students a storyboard** ✂✎ A storyboard (→ storyboard) could help students in their game design process, the template refers to the "Shape of a game" • Ask students to give their game a name • Let them tell a story	**Playing – Game Design**	☐
	2.2.2 Classify the game: genre/theme/goal * ✎ • **Choose a genre**: ○ Action (platform/jump'n'run, shooter) ○ Adventure (RPG, text adventure/storytelling) ○ Puzzle (skill game) ○ Quiz ○ Simulation (racing, real-life)		☐

- o Strategy
- **Choose a theme**:
  - Criminal/detective stories,
  - Science fiction, fantasy, comic
  - Romance
  - Nature, animals, sports
  - Future, space
  - Realistic
  - Horror, etc.
- **Choose a goal**:
  - Capture/destroy/avoid e.g., items or opponents
  - Territorial/knowledge acquisition, collection, e.g., items
  - Solve a puzzle or a crime
  - Chase/racing/escape something or somebody
  - Spatial alignment: positioning of elements
  - Build a character, resources
  - Negation of another goal: games end if the play act against the rules
  - No goal (e.g., storytelling, retelling, animations)

2.2.3 Who is the "star" in the game? * 🖉

- Main characters, e.g., animals, fantasy figures, man/woman, boy/girl, items, transport, food, etc. ☐
- Side characters
- Name all the characters to promote ownership
- Background (i.e., theme)
- Interactions between characters and their level of control

2.2.4 Bring the games to LIFE (use MDA) *
Mechanics → Dynamics

- Points/rewards: e.g., earning points/currency to levelling up (reward completion of activities) or for a high-score list ☐
- Status/levels: thresholds or milestones that a player must achieve in the progression.
- Challenges/achievements: tasks or actions users have to perform to be awarded
- Virtual goods/self-expression: non-physical, intangible objects the user can, for example, exchange in virtual shops to customize their avatar
- Leaderboards/competition: scores and rankings of users relative to others (e.g., high-score list)
- Notifications: provide feedback for the user
- Timer: set a time limit for actions

Aesthetics: provide visual, audio, and fantasy elements

- Sensation: create something completely unfamiliar
- Fantasy: build imaginary worlds
- Narrative: tell a story
- Challenge: to master something
- Fellowship: the player is part of a community
- Discovery: the players need to explore
- Expression: use individual creativity

2.2.5 Get the games in shape! "Ceremony" 🖉

- Title screen: name of the game
- Introduction screen: explain the goals and rules (mechanics) of the game
- Game screen(s): 1-n levels ☐
- End screen: game over or win screen

(2.3) Let students be creative and express themselves	**2.3.1 Foster students' sense of ownership - It's their game! ***
	• Also within templates/frameworks!
	• Edit/change, customize and personalize: assets, characters, looks, backgrounds and screens (shape of a game)
	• Add sounds, record media
	• Suggestions:
	    o Use art lesson for design session
	    o Have group members already started to code? No problem: Let them *change* roles after a while!
		Creativity – Freedom of Choice	☐

	Stage 3 – Coding		

(3.1) Now let's start coding!	**3.1.1 Tinkering activities: First … on paper! ✄**
	• Pseudocode: students should think of commands, variables, etc. they will need for their games
	• Hands-on/bricks on paper: print out the bricks, students add them to their objects
	    o Where to place the objects?
	    o Which size they are?
	    o How I will control my objects?
	    o How and who should interact/communicate with each other?
	    o How will I use MDA in my game?
	    o Where to define my variables?
		Coding – Personal Experiences	☐

	3.1.2 Ready…Set…Code! * • Students should try it out and see what happen, e.g., If something does not work like expected: change it • Consider failure as part of the learning process • Do not show/explain all at once: break down the content into sub-goals for every units	**Engagement – Collaboration**	☐

	3.1.3 Repeat, focus, and foster collaboration *
	At the beginning of every unit:
	• Let students repeat what happened in the last unit and present sub-goals for today's unit.
	• Ask: What was difficult? What was easy? Open questions?
	• Observe the teamwork: enable students to assume different identities and roles (leader, designer, programmer, etc.).
	• Build confidence:
	    o Praise students, provide confirmation
	    o Celebrate "Aha!-effects"
	    o Provide recognition of work done
	    o Balance extrinsic and intrinsic motivators
	• Support collaboration and communication during the whole game production process
	• Foster originality and self-expression
	    o What else can you add to the game?
	    o Are you satisfied with your game?
	    o Is anything missing?
	    o Is there some room for improvement?
	    → Students feel pride/self-efficiency!
	• Check the state of the work:
	    o Who needs more time?
	    o Provide extra tasks for the faster ones.
		Creativity – Freedom of Choice	☐

<table>
<tr><td>(3.2) Don't forget the gender</td><td>3.2.1 Be gender-sensitive/aware *
• Be sure your learning materials are free of gender stereotypes (example games, learning goals, templates/frameworks).
• Use a gender sensitive language, e.g., do not foster male masculinity in tech (e.g., only refer to a technician as HE), consider that language forms pictures; so make women visible and audible, use both definitions (e.g., in German) or more neutral forms if they exist.
• Use gender sensible language and imaginary for slides, material, and examples.
• Praise students the right way: not for spent effort/time, but for their knowledge.
• Provide a stress-free and anxiety-free working environment by considering different skill levels or preferences.
• Ensure a competition-free environment.
• Observe groups/individuals: who is engaged, who asks, and who is holding back.
• Support (girls) to pursue and persist in technology</td><td>Engagement – Collaboration</td><td>☐</td></tr>
<tr><td colspan="4">Stage 4: The Closing</td></tr>
<tr><td>(4.1) Enable recognition of the student's progress by peers, teachers, and parents</td><td>4.1.1: Allow students to present their games in public to provide a sense of ownership & pride
(voluntary/mandatory)
• In front of their peers (during the last unit)
• At events (e.g., open house days, final event)
• Sharing (i.e. through a public forum)
• Recap session / ask questions:
    o Who will program at home? Tell his/her friends?
    o Highlights/problems, etc.

4.1.2: Make a short quiz ✄
• Discuss the questions at the beginning of the unit with the whole class
• Define easy questions, e.g., single choice questions
• No teamwork, no grading
• Discuss the questions after the quiz

4.1.3: Evaluate submitted programs 🔔✄
Assessment of:
• Confirmation of achievement of the learning goal(s)
• Use of game design elements
• Program structure (e.g., code statistics, finished program)
(→ assessment template sheet)</td><td>Engagement – Collaboration

Coding – Structure</td><td>☐

☐

☐</td></tr>
</table>

Hints: per unit at 45 minutes (1) (2) (3): steps (a) (b) (c): choose
Legend: * gender sensitive 🔔 only for schools ✄ optional ✎ only if coding from scratch

A.5. PECC Storyboard Template

Storyboard

1) Title scene	2) Introduction scene
3) Game scene(s)	4) End scene

Name of your game:	
Main character:	
Gameplay (what is your game about?)	
Genre/theme **Genre**: Action (platform, jump'n'run, shooter) Adventure (RPG, storytelling) Puzzle (skill game), Strategy Quiz, Simulation (racing) **Theme**: Horror, Romance, Science Fiction, Space, Sport, Future, Nature, Fantasy	
Goal Capture, destroy, avoid, solve, chase, racing. escape, build, etc.	
Mechanics/Dynamics Points, level, challenges, inventory, high-score, timer	

A.6. PECC Assessment Template

PECC-Assessment template

PLAYING/CREATIVITY (DESIGN ELEMENTS)

Genre 1:action, 2:adventure, 3:puzzle, 4:quiz, 5: simulation/racing, 6: strategy/skill, 7:platform/jump'n'run, 8: action shooter 9:RPG adventure, 10:text adventure/storytelling	
Theme 1:crime/detective, 2:science fiction, 3:fantasy, 4:romance, 5:sports, 6:nature, 7: space, 8:realistic, 9:horror, 10:comic	
Goal 1:capture/avoid/destroy, 2: territorial, 3: acquisition, 4: collection, 5: solve, 6: chase/racing/escape, 7: spatial alignment, 8: build, 9: negation of another goal, 10: no goal/animation	
Shape of a game (yes/no) title scene, introduction scene, game scene(s), end scene	
Level of control e.g., sensors, keys, buttons,…	
Visual design 1:handmade artwork, 2:painting app/tool, 3:internet, 4:media library, 5:own photograph, 6: photoshop	
Sound design 1:own records, 2:internet, 3: media library	
Main character	
Others e.g., restart button, timer progress bar, difficult levels,…	
Deviations from the original storyboard/idea (yes/no)	

GAME ELEMENTS (MDA) x=used

	Planned game elements	**Implemented game elements**	**Postponed game elements due resource limits**	**Added game elements through game development**
Levels/reward				
Points/status				
Challenges/achievements				
Virtual goods				
Self-expression				
Leaderboard/competition				
Sensation				
Fantasy				
Narrative				
Fellowship				
Discovery				
Expression				
Feedback (for the player)				

TEAMWORK

Group composition SG: small group, PW: pair work, I: individual	
Role/responsibility within the group	

ENGAGEMENT / CODING

Scale 1-4 (1=lowest, 4=highest)

Collaboration Observations regarding collaboration, support, and interaction in classes with other students while coding.	
Teacher intervention (positive intervention/mentoring) Taking into account the interest shown in class as well as the questions and interactions of the student with the teacher.	
Match of (learning) goals & gaming objectives Achievement of the pre-defined learning goals (defined by the teacher). Adherence to the academic theme or content that needs to be learned by the student.	
Match of gaming objectives Game's appearance and used MDAs, and game design aspects.	
Game originality Originality of the game, or how the academic concept "is gamified".	
Complete/sequence /flow/story structure How the concept and its different areas/parts are represented in the game (use of the "Shape of a game")	
Work defense Presentation of the game, "defend" the performed work, and answer to questions.	

CODING

Use of template/framework (yes/no)	
Number of scripts/classes	
Number of bricks/functions	
(…)	
Number of objects	
Number of looks	
Number of sounds	
Number of global variables	
Number of local variables	

A.7. Lesson Plan — Example of a PECC Activity for Female Students

Example of a PECC Workshop (Lesson plan)

For Grade 9

Unit 1

Introduction to Pocket Code/Game design

Structure of the unit: introduction, presentation, bricks on paper activity, game design

Lesson at 50 minutes: 1-2 units

Unit 2

Coding

Structure of the unit: bricks on paper activity, coding

Lesson at 50 minutes: 1-2 units

Unit 3

Presentation and closing

Structure of the unit: coding, presentation, discussion

Lesson at 50 minutes: 1-2 units

Unit 1

Introduction to Pocket Code

Structure of the unit: introduction, presentation, brick-on paper activity, game design **Lesson at 50 minutes**: 1-2 units

Checklist

Power point presentation with examples, task explanation, and learning goal	☐
For the bricks on paper activity per student: set of bricks, 3 planets, 1 background, 1 character, 1 motion type, coordination system, tap, scissor, A3 paper	☐
Informed consent of parents	☐
List of usernames/passwords	☐
At the end of the lesson: collect the produced templates	☐
At the end of the lesson: preinstall the "Shape of a Game" on the devices[1]	☐

Timetable

5 minutes	Intro
15 minutes	Warm-up
15 minutes	Presentation
15 minutes	Explanations to the task
45 minutes	Bricks on paper activity
5 minutes	Closing
2 **Units** à 50 minutes	

Learning Goal(s)

Students can apply the concepts of loops/broadcast messages and explain for what they are used for.

Students can use bricks (different actions) and put them together in order to create a syntactically and semantically correct pseudocode on paper.

Result

All students start with the bricks on paper activity (coordination system) and choose their objects (own assets).

[1] Example: https://share.catrob.at/pocketcode/program/48517

Sequence:

Intro

Duration: 5 minutes

- What to expect (structure of the units, agenda)
- Collect informed consent
- Assign credentials to students

Warm-up

Duration: 15 minutes

Discussion:

- Which technical professions do you know?
- Which study/training do you need to acquire technical skills?
- What does a computer scientist do? Do you know people who are working in those fields?
- Who already has experience in coding? Which tools did you use for coding?
- What is coding? What is an algorithm?
- Which programming languages do you know?
- What kind of games do you play?
- What makes you play games?

Talk about role models, famous female IT specialists from the history and the IT profession in general.

Introduction to Pocket Code

Duration: 15 minutes

Presentation & collaborative program creation

- Slide 1: show the app Pocket Code + how to download
- Switch to live-demo 1
  - Show the menu/UI
  - Show "Explore" (community sharing platform), programs
  - Create a new program (starter program): show scripts, looks, sounds and brick categories
  - Ask students for the next step while programming
  - Show a program that already has two objects with two looks [2] (ask: "What is an object?)
    - Add a look
    - Add a brick „When tapped" and switch the look "Next look"
- Slide 2: explain loops
- Slide 3: explain broadcast messages
  - Switch to live-demo 2/starter program
  - Add a "Forever loop" brick and a "Wait" brick to program (to create an animation)

[2] https://share.catrob.at/pocketcode/program/41332

- Add “broadcast messages” to the program (“When tapped”, “Send to all”, second object: “When I receive” “Say Hello!”)

Explanation of the task

Duration: 15 minutes

“Your object wants to learn something about the planets and goes on a journey to outer space. The planets tell him/her some facts about themselves if the object touches them.”

Discuss the task: What concepts are needed?

- Slide 4: explain the task and learning goal (for today)
  1. Bricks on paper activity
  2. Take one background
  3. Choose one object
  4. Choose 3 planets
  5. Take a set of bricks
  6. Take “level of control” for the main character (inclination sensor, “When tapped” proberty) – allocated randomly
- Live demo 3: show the different kinds of movement with a program

Hint: Everybody should create his own program on paper. Students can work together in regard to their assigned sensor and can create their own assets (characters, planets).

Bricks on paper activity (hands-on) / Pseudocode

Duration: 45 minutes

- Slide 5 with hints:
  - Name your object.
  - Define the size of your object.
  - Where should you place your object?
    - Ask: What is a coordination system?
      (→ coordination system)
  - What happens if the object touches the planets?
  - Which object needs to communicate with the others and when?
  - What do the planets have to say?
- Slide 6: show a slide of how it could look (the brick on paper template)
  - Ask: “What is a pseudocode?”

Closing

Duration: 5 minutes

Summarize and recap.

Unit 2

Coding

Structure of the unit: bricks on paper activity, coding Lesson at 50 minutes: **1-2 units**

Checklist

Distribute tablets/smartphones (note down which students use which tablets)	☐
Distribute the handmade templates/collect them again at the end of the lesson	☐

Timetable

5 minutes	Repetition
35 minutes	Finish bricks on paper activity
10 minutes	Research facts to planets (internet)
05 minutes	How to start?
40 minutes	Coding
5 minutes	Upload
2 Units at 50 minutes	

Learning Goal(s)

Students create their game according to their self-made template. Students use the app Pocket Code, add bricks (actions), objects, and looks to their games, do research (research facts to planets), and try to solve these tasks alone or in teams.

Result

All students finish the bricks on paper activity; they collect facts about the planets (3 facts per planet) and all start to code in Pocket Code.

Sequence:

Repetition

Duration: 5 minutes

Discuss:

- What did we do the last time?
- How many of you already finished with X or X (last sub-goals)? Who needs more time?
- What was difficult? What was easy? Open questions?

Finish bricks on paper activity

Duration: 35 minutes

Foster teamwork and provide support.

Research facts to planets (internet research)

Duration: 10 minutes

- Allocate the tablets.
- Students search for three facts to include about each planet (internet research).

How to start?

Duration: 5 minutes

- Slide 7: Explain the “Shape of a Game”.
- Life demo 3: Show how to add new scenes and rename the scene.

Coding

Duration : 40 minutes

Start with your games.

Provide hints: add each brick one by one and execute the program to see what happens. Test often and adjust if something is not right

Foster collaboration:

- Observe the teamwork.
- Build confidence: praise students, provide recognition of work done.
- Balance extrinsic and intrinsic motivators.
- Foster originality and self-expression.
- Be gender aware!

Upload and Closing

Duration: 5 minutes

Upload programs and collect templates.

Unit 3

Presentation and Closing

Structure of the unit: coding, presentation, discussion Lesson at 50 minutes: **1-2 units**

Checklist

Prepare quiz	☐
Program upload	☐

Timetable

5 minutes	Repetition
70 minutes	Coding, extra tasks, uploading
10 minutes	Students present
10 minutes	Quiz
5 minutes	Discussion and closing
2 Units at 50 minutes	

Learning Goal(s)

Students code their own games and integrate the "Shape of a game" in their games. They can explain the different parts and concepts used.

Result

Students can present their programs to their peers and explain different concepts. They upload their programs to the community website.

Sequence

Repetition

Duration : 5 minutes

Discuss:

- What did we do the last time?
- How many of you already finished with X? Who needs more time?
- What was difficult? What was easy? Open questions?
- Ask questions you are going to ask at the end of the unit during the quiz.

Finish programs (coding) and upload the game

Duration: 70 minutes

Students who finished earlier can work on extra tasks:

- Include the "Shape of the Game" (use of variables)
    - Ask: When does the game start/end? What are variables?
- Add a quiz at the end with questions about the planets. (Use of "Ask" brick)..
- Help your colleagues.

Upload the games.

Presentation

Duration: 10 minutes

Two volunteers present their games in front of their peers (one who used the inclination sensor and one who used "When tapped"-property, one girl/one boy).

Sequence:

- Present the game (play).
- How did you manage the movement of the object?
- How did you add the end screen?

Quiz

Duration: 10 minutes

Possible questions:

- What happens when you tap on the play button?
- Each object consists of: (a, b, c).
- Which block categories can you remember?
- Which loops did you use or which ones can you remember?
- What do you need for objects to communicate with each other?

Discussion and Closing

Duration: 5 minutes

Discuss in plenum:

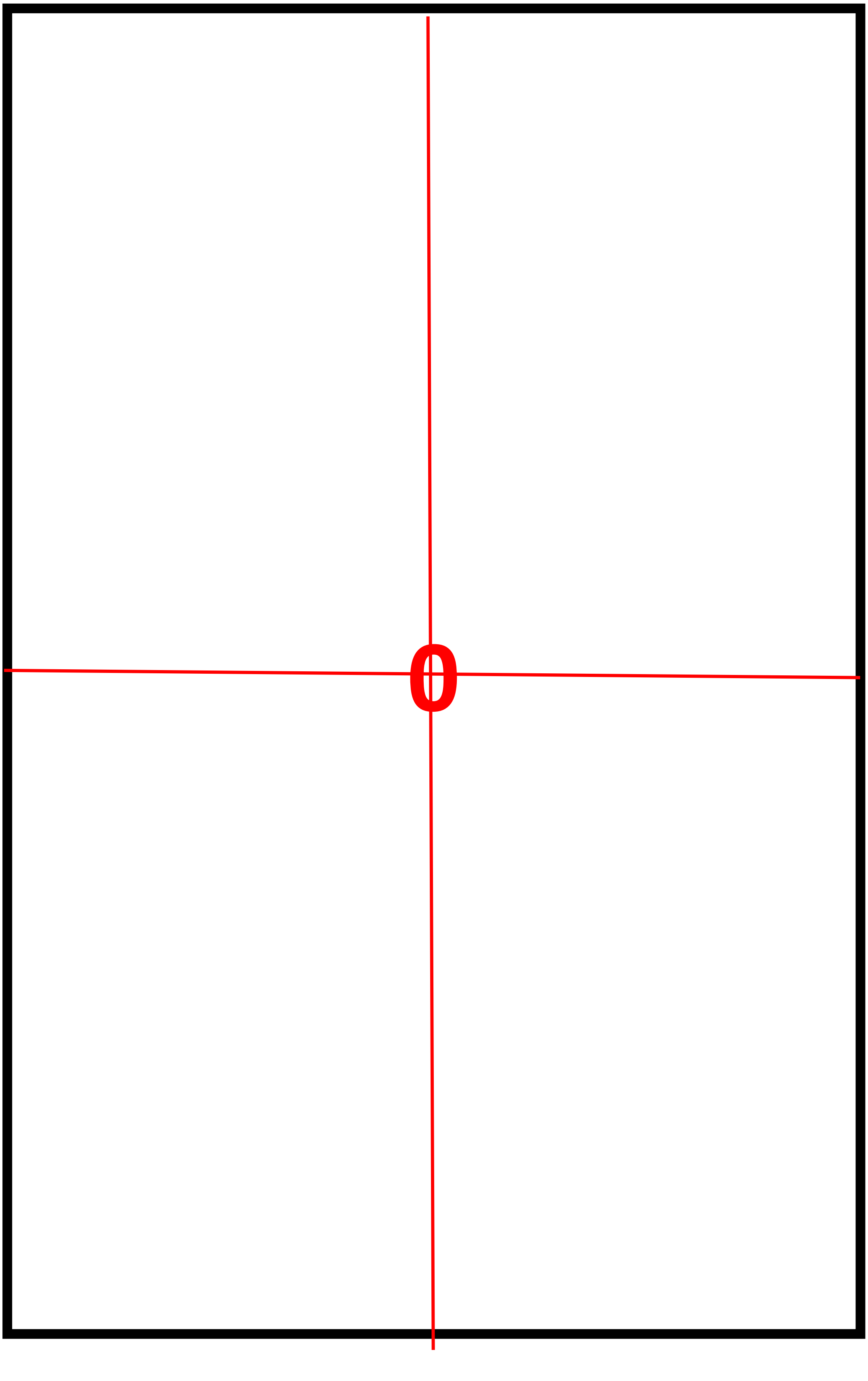
- The answers of the quiz

- Who is planning to continue coding with Pocket Code? (reasons for/against)
- What did you like the best/the worst? Recommendations?
- Do you want to use the app in other/in this courses again?

A.8. Coordinate System

Your device screen

Add the size of you screen. The coordinate system is visible if you pause the game during playing (back button) and tap on “Axis on”.



A.9. IT Glossary for Facilitators

Used sources:

- https://learning.linkedin.com/blog/tech-tips/coding-de-coded--18-terms-new-programmers-should-know
- https://techterms.com/
- https://wiki.scratch.mit.edu

Broadcast Message

Broadcasts are messages that allow users to send messages to all at once or to a specific group. Examples are newsletters, sales offers, or seasonal messages. In Pocket Code messages are used to communicate among objects.

Conditional

Conditions are there to process decisions, e.g., when you have to choose which way to proceed. They are evaluated to either be TRUE or FALSE. Furthermore, multiple conditions can be combined into a single condition with the same final value.

Data type

They represent a specific kind of data item, defined by the values it can accept, the programming language, and/or the operations that can be performed on it. For example integers (simple numbers), floating numbers (decimal-based numbers), Boolean values (TRUE or FALSE) and strings (a text or words), etc.

Expression

An expression is a mathematical statement that comprises of variables, numbers, and operations, which results into a specific value. For example “(x + y)/ 100”.

IDE

This is short for Integrated Development Environment and means a development tool (or collection of tools) used to develop particular languages. They can include code editors, compilers, debuggers, and other utilities. For example, Microsoft’s Visual Studio, and Java IDEs like IntelliJ, NetBeans, or Eclipse.

Loop

A sequence of instructions that are continually repeated (looped) until a certain condition is reached. For example repeat until, or forever, etc.

Object

In object-oriented programming languages, an object has a name, a state (defined by his values) and a behavior (defined by his actions). For example a cat has states such color, and hunger, and her behavior is meowing, hissing, or clinging.

Parameter

A parameter or "argument" in programming is a value that is set into a function. Functions can usually have multiple parameters.

Pseudocode

A pseudocode is like a rough draft of the code, which shows an outline of the intent and flow of a program before writing and debugging the actual code.

Variable

In math, a variable is a symbol or letter that represents a value. In programming, a variable is a named identifier with a value that changes throughout the execution of the code. Once a variable is defined in the program, the value can be changed when needed. Examples are variables for a score or a time.