Bridging the Gap: A Computer Science Pre-MOOC for First Semester Students

Bernadette Spieler[1], Maria Grandl[2], Martin Ebner[2] and Wolfgang Slany[3]
[1]University of Hildesheim, Institute for Mathematics and Applied Informatics, Hildesheim, Germany
[2]Graz University of Technology, Institute of Interactive Systems and Data Science, Graz, Austria
[3]Graz University of Technology, Institute for Software Technology, Graz, Austria
bernadette.spieler@uni-hildesheim.de
maria.grandl@tugraz.at
martin.ebner@tugraz.at
wolfgang.slany@tugraz.at
DOI: 10.34190/EJEL.20.18.3.004

Abstract: Knowledge in Computer Science (CS) is essential, and companies have increased their demands for CS professionals. Despite this, many jobs remain vacant. Furthermore, computational thinking (CT) skills are required in all contexts of problem solving. A further serious problem arises from the gender disparity in technology related fields. Even if tech companies want to hire women in technology, the number of women who enter these fields is remarkably low. In high schools with no technical focus, most teenagers acquire only low-level skills in CS. The consequences are misleading preconceptions about the fundamental ideas of CS and stereotype-based expectations. Consequently, many teenagers exclude computing from their career path. In this paper, two promising concepts to overcome these challenges are presented. In 2018, a voluntary gamified lecture "Design your own app", held at the University of Graz for students of all degree programs, was introduced. The course attracted over 200 students and received positive evaluations. This led to the second concept. In January 2019, a MOOC (Massive Open Online Course) with the title "Get FIT in Computer Science" was designed and launched in August 2019 on the platform iMooX.at with the goal to provide a basic introduction to different concepts of CS, including programming and the application of game design strategies. The MOOC was accompanied by an offline lecture, following the principles of flipped classroom and inverse blended learning. For evaluation purposes, we collected data at three stages: 1) during the MOOC, 2) during the offline lecture, and 3) two months after the lecture. The results showed that the MOOC framework was a promising approach to support and motivate at least a certain group of first-semester students, especially those who had no prior knowledge in CS.

Keywords: computer science education, digital literacy, technology enhanced learning, MOOC, flipped classroom, Pocket Code

1. Introduction

IT specialists are needed worldwide in order for businesses to be competitive in science and technology. Thus, there is a growing demand for IT professionals (Cuff, 2015). On the one hand, the enrolment of students in Computer Science (CS) programs in European countries has slightly increased, according to Informatics Europe (2015). On the other hand, degree courses in CS have very high drop-out rates (European Commission, 2015). There are many reasons why students drop out of university, but this report points out that students who have no previous knowledge in CS are more likely to drop out in comparison to students who already have programming experience and a basic understanding of the important concepts of CS. In addition, data from European Statistics Eurostat (2019) confirm a low percentage of female students in studies related to CS. In Austria, the percentage is only about 14%, which is even lower than the EU average (17%), but higher than the share of women in ICT studies in Luxembourg (10%), Belgium (8%) and in the Netherlands (6%). Female students, who decide to enter the field of ICT or CS, are then confronted with various challenges and prejudices. In STEM (Science, Technology, Engineering and Math) subjects in Austria, all male and female students in 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 non-STEM 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 (Dormayr and Winkler, 2016). With regard to success rates and gender, the only exception is the field of CS: Here, women have a 10%-points lower success rate than men. One explanation of this gender related success rate is the differences in school education. Overall, women in STEM studies more often have a degree from an upper secondary school providing

Reference this paper: Spieler, B., et al., 2020. Bridging the Gap: A Computer Science Pre-MOOC for First Semester Students. *The Electronic Journal of e-Learning,* 18(3), pp. 248-260, available online at www.ejel.org

a general education (women 66% vs. men 50%) and rarely completed their secondary education at a technical college (women 8% vs. men 37%).

The percentage of female students at Graz University of Technology (TU Graz) in Austria for the winter term 2017/2018 reaches just 23.87%. This percentage increases slightly every year, 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 12.55%, without significant changes over the last 20 years. The gender distribution at the TU Graz in the different levels continues to show inequality. With 76.13% male students, female students are in the clear minority.

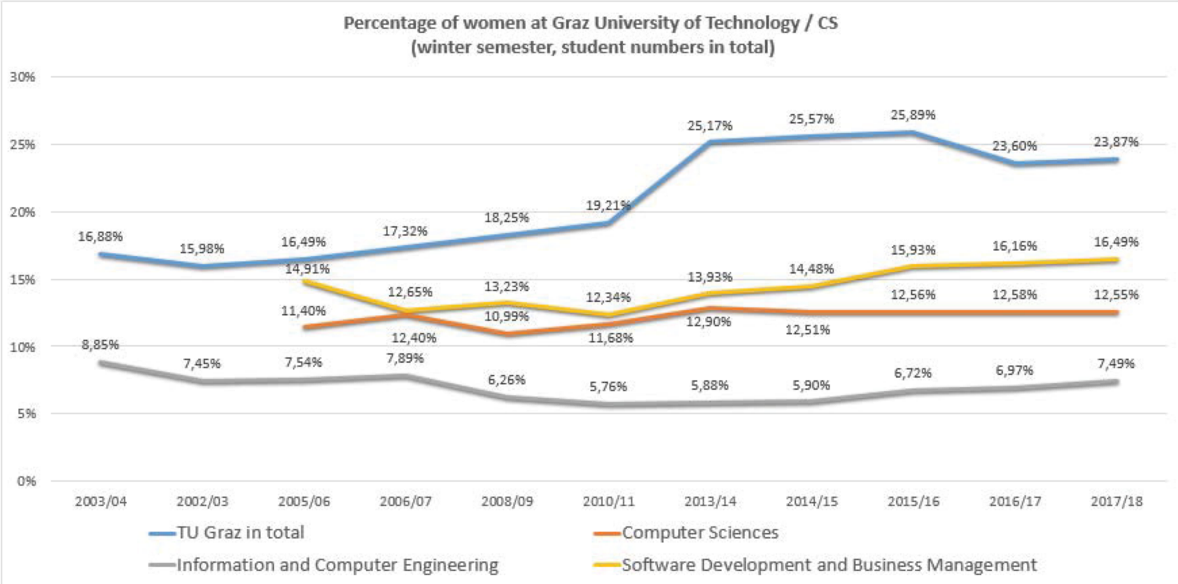

Figure 1: Percentage of CS degree programs at TU Graz (winter term of 2003/04 to 2017/18)
https://online.tugraz.at/tugonline/Studierendenstatistik.html

This situation is similar to other developed countries. Reasons why female teenagers decide not to choose CS as a major are diverse. Literature and researchers argue that existing stereotypes, preconceptions about the field of CS, 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. These important gender differences especially in interest, self-efficacy or sense of belonging are part of the authors' previous work (Spieler, Oates- Induchovà, Slany, in press). These issues need to be considered when thinking of alternative approaches to increase female achievement in this field (Cheryan et al., 2013). In addition to this, there is a major lack of exposure to CS at schools all over Europe (CECE, 2017).

The aim of the paper is twofold: First, the development of a MOOC (Massive Open Online Course) and the implementation of a corresponding offline lecture at TU Graz, providing a general introduction to CS, are described. The MOOC and the lecture were designed to help all first semester students to gain knowledge in CS and programming. Second, the results of the online course and the offline lecture were compared with the answers of a post questionnaire, done by CS first semester students in December 2019 to see if the topics covered by the MOOC were helpful. For those who did not participate in MOOC, we wanted to know which topics would help them in their first semester courses. This paper investigates the following research question: How can a MOOC in the form of a preparatory course help students and especially women with different background knowledge in technology to acquire CS and programming skills?

This paper is organized as follows: First, relevant concepts, methods and arguments for the design of the pre-MOOC are presented from a theoretical point of view. Based on the theoretical framework, the authors describe the research method, followed by a discussion of the final results related to the MOOC, the offline lecture and additional questionnaires. The last section concludes this paper and describes the author's future work.

2. Related Literature

A New York Times article by Singer (2019) stated that a growing number of students with no experiences in CS feel motivated to learn about coding but are not willing to study CS as a major. Offering courses at the university level that focus on a general introduction to CS and, more precisely, on computational thinking skills instead of digital literacy, may lead to better career prospects and a better understanding of the importance of CS in nearly all domains. In this section, related literature referring to the content, format, didactics and pedagogics of a CS introduction course is presented.

3. Digital Literacy versus Computer Science

A critical approach to new technologies requires a general understanding of the logical and technical aspects behind them. However, European school systems are mostly concerned with Digital Literacy and Information and Communication Technology (ICT) as a supporting context-free medium/technology to enhance learning." (CECE, 2017). CS is more than just using the computer as a tool but a "distinct scientific discipline" that plays an insignificant and insufficient role in the school curricula of most European school systems (ibid.). Consequently, CS remains a great unknown for many pupils worldwide. Computational Thinking (CT) is considered an essential skill in the 21st century. According to J. Wing, it "represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use." (Wing, 2006). Under certain conditions, coding activities are one way to train CT. Many countries around the world, including Austria and Germany, organize an annual Bebras contest that usually takes place in November (Brebas Challenge UK: http://www.bebras.uk/). The aim of this contest is to help pupils of different age groups to develop computational thinking skills and to boost interest in CS without the actual use of a computer. Furthermore, CS unplugged activities are a way to teach fundamental principles of CS playfully by using paper, strings, crayons, and movement (Brackmann et al., 2017).

4. Technology Enhanced Learning and Massive Open Online Courses (MOOCs)

ELearning concepts, online courses, or MOOCs, in particular, are a perfect way to support distance education or lifelong learning. The rise of MOOCs can be dated back to the so-called "year of the MOOCs" in 2012 (Pappana, 2012). Since then, more and more higher education institutions began to produce so-called xMOOCs (Carson and Schmid, 2012), by using large MOOC platforms like Udacity, edX, or Coursera to publish their video-based courses or lectures. It has become apparent that those MOOCs helped to reach a broad public and to introduce new didactic approaches (Ebner and Schön, 2019). However, the most well-known problems of MOOCs are the high drop-out rate (Khalil and Ebner, 2014) and the difficulty in certifying courses in the context of a higher education institution (Kopp and Ebner, 2017). In 2014, TU Graz and the University of Graz founded the first and currently only MOOC-platform in Austria, called iMooX (Khalil and Ebner, 2016). Following the idea of open education, each single learning object within a course on iMooX is published with an open license and can be identified as an Open Educational Resource (OER). In recent years, more than 60 MOOCs have been offered on iMooX.at, addressing people from different educational sectors with different interests. In the context of this research work, one of the most relevant courses was the so-called "Mathe-FIT-MOOC" (Get-FIT-In-Math-MOOC). This MOOC is combined with an offline lecture and aims to act as a preparatory course to bridge the knowledge gap in Math between school and university level.

5. The Pre-MOOC Principle: From Flipping the Classroom to Inverse Blended Learning

Following the research study and the design principles of Ebner et al. (2020) there are seven learning and teaching scenarios to use MOOCs in higher education - from a very traditional, pure online form to a mix between face-to-face and online education. The MOOC "Get-FIT-In-Math" follows the concept of a pre-MOOC, meaning that the MOOC starts and ends before the face-to-face education (Ebner et. al, 2020, p. 79). This type of MOOC is used when students (or other learners) need to have some prior knowledge or if there are simply some other restrictions in time or place.

The underlying didactical design is a mix between the well-known flipped or inverted classroom and the concept of Inverse Blended Learning. Flipping the classroom is the idea that students can study the content of a lecture at home (mostly by watching videos of the instructions of the lecturers) and do their "homework" or exercises in class (Li et al., 2015). Inverse Blended Learning (IBL) is following the idea that education and therefore learning is a social process which must be done by the learners themselves - people interacting with people; the teacher interacting with the learners; learners interacting with the teacher and just as important, the learner interacting with other learners (Price and Lapham, 2003). With other words, we have to ensure and foster interaction to strengthen the learning process instead of rather consuming content. MOOCs traditionally lack interaction, which leads to a high drop-out rates in those courses (Khalil and Ebner, 2014). IBL aims to bring the pure online course back to face-to-face education. For example, different forms of learning meetings, in parallel to the MOOC, can be arranged to ensure discussions between learners or give them the possibility to exchange with instructors in case of any questions. Both, the principles of flipped classroom as well as IBL increase the retention rate in MOOCs (Ebner et al., 2017).

6. Game Development Based Learning (GDBL) Challenges and Tools

Game Development Based Learning (GDBL) challenges are popular in introductory programming courses at all educational levels and a popular strategy to motivate and engage students in CS topics (Wu and Wang, 2012). The active involvement as designers and producers of their own games promotes creativity, problem-solving, and critical thinking skills (Vos, van der Meijden and Denessen, 2011; Ya-Ting et al., 2013). Relevant knowledge in CS should help (young) people to consider new career paths that can ultimately lead to a fast employment and well-paid jobs (European Commission, 2016). New technologies and tools formed the ways of learning and teaching in the 21st century. For example, web-based technologies like HTML5, as well as an increase in the number of modern smartphones and tablets, opened up new ways for innovative coding concepts (Kahn, 2017). Game Development-based approaches could be easily applied with block-based visual oriented programming tools, like Scratch (https://scratch.mit.edu/) and support novices in their first programming steps (Tumlin, 2017). In 2010, the Catrobat team (https://catrobat.org) at TU Graz developed an educational app that allows the creation of games, stories, animations, and many types of other apps directly on smartphones or tablets, see Figure 2.

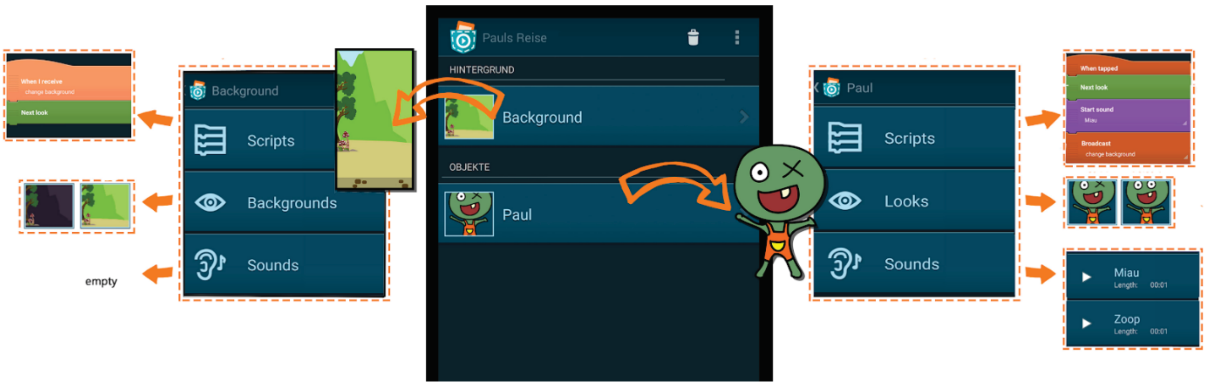

Figure 2: Pocket Code's User Interface. Free on Google Play and iTunes.

7. Diversity in Computer Science

The introduction section stated the problem: The percentages of female university students in ICT fields in Austria currently varies between 4% to 20% and the drop-out rate is much higher among female students than among male students (Binder et al., 2017). This leads to a small number of female professionals in CS and consequently to missing role models and male-dominated development teams.

Many researchers conclude that there is no need to "fix the women", but to fix the system (e.g., Master, Sapna, and Meltzoff, 2016; Alvardo, Cao, and Minnes, 2017). Predominant stereotypes, and missing role models (Young et al., 2013), as well as other social and cultural factors (e.g., gender role socialization, peer groups), expose technology as male-dominated field. Girls' lower sense of belonging corresponds with the feeling of "Lack of Fit" or "the Sense of Not Fitting In" with CS stereotypes. This occurs when female CS students feel that they do not receive help, question their ability in CS, or feel intimidated by others. If the profession does not fit the "traditional gender model", one is not as likely to pursue or feel discriminated against by someone who does. To be socially connected and respected is a strong initial motivator (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, 2016). 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 (Cheryan, 2012;

Master, Sapna and Meltzoff, 2016). All of these factors make young women question whether their abilities and interests are harmonious with their selected field.

8. Methods

To explore the MOOC concept as a supportive tool for female first semester students to gain knowledge in CS and programming, we developed a new CS pre-MOOC. This MOOC, with the title "Get FIT in Computer Science", is based on the experiences and feedback gained from the obligatory lecture "Design your own App" and followed the principle of the "Get FIT in Math" MOOC. "Get FIT in CS" was launched on August 5th, 2019 on the iMooX-platform and was offered in combination with an offline lecture at TU Graz.

9. Research Design

The voluntary lecture "Design your own App" was introduced in the summer terms of 2018 and 2019 for all students at the University of Graz. A total of 202 students from diverse degree programs attended the first lecture. A majority of students (43.82%) came from three degree programs: 17.98% from teacher training studies (e.g., studying a language, maths, geography, physics), 15.73% from Business Management and Economics, and 10.11% from CS. A percentage of 56.18% came from 120 different degree programs, including psychology, legal sciences, molecular biology, sociology, history, chemistry, or philosophy. This variety of students was impressive. The goal of the lecture was to 1) stimulate computational thinking skills, by using logical puzzles (e.g., Beaver challenges), and 2) foster participation and creative thinking via the game-making and coding tool Pocket Code.

The newly designed MOOC "Get FIT in CS" consists of seven lectures, which are presented in Figure 3. It is still possible to register for the course by following the link: https://imoox.at/mooc/go/Info-Fit19. The MOOC provides a general introduction to the field of CS and programming. Besides videos, the MOOC includes interactive exercises (e.g., gap text, timelines) and programming tasks. Students who participated in the MOOC were supposed to apply game design strategies by using Pocket Code and while learning the basics of programming with Python and Java. Every lecture included exercises (e.g. to write pseudocode, execute assembler commands, solve tasks of the "Bebras challenges"), and self-assessment questions are presented in a multiple-choice format. Furthermore, students were encouraged to post and discuss their results in the corresponding online forums and received feedback from the course leaders and/or the other participants. Participants who completed all seven lectures had to fill out a final survey and receive a certificate.

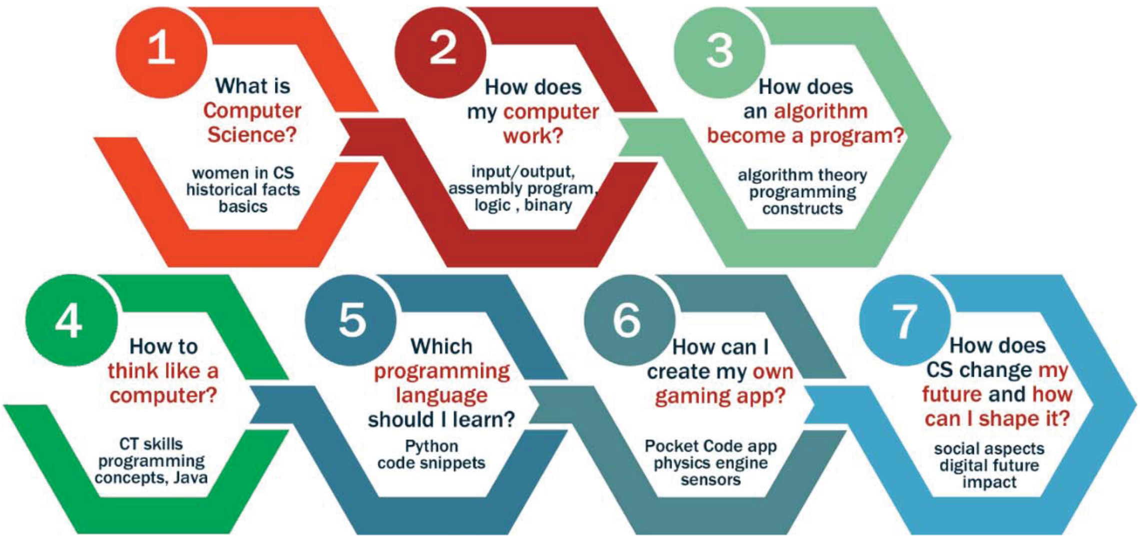

Figure 3: The MOOC "Get FIT in CS" consists of seven lectures. The content of each lecture is represented by a central question

This MOOC was offered in combination with an offline lecture at TU Graz. The identically named lecture took place on five consecutive days, between the 18th and the 25th of September 2019, and lasted two hours per day. The lecture followed the structure of the MOOC and aimed to repeat the main topics of the MOOC, practice computational thinking, and prepare students for upcoming challenges in their degree program at the university. For most of the students this lecture was their first lecture at TU Graz. Thus, the lecturer also focuses on organizational details (e.g. WLAN access, learning management system). Students received a grade and 1.5 ECTS if they completed the MOOC by the 25th of September, attended 80% of the offline lecture, submitted homework tasks (e.g., a Python program), and submitted and peer reviewed a Pocket Code game. During the lecture, exercises were discussed with the students in an interactive way, ranging from a CS-unplugged human binary counter to a demonstrative explanation of the bubble sort algorithm and tasks of the "Bebras" initiative (see Section 2.1).

To attract more female students, we tried to counteract stereotypes and clarify expectations. First of all, nearly all videos in the MOOC show a female speaker (the lecturers). Secondly, in lecture one, a video was created to portray famous female (computer) scientists. The organizers also decided to use a game design approach to highlight how programming is a highly creative process. Completed games could be presented on a voluntary basis on the last day of the lecture. During the offline lecture, both female lecturers used gender-sensible language and talked about female achievements, which received many positive reactions. In order to elaborate the process of asking questions during the lecture, an audience response system (ARS) was used.

10. Data Collection Process

Data was collected in three phases: 1) online questionnaires after students completed the MOOC, 2) during the lecture with the help of audience response systems (ARS), lecture evaluations, and submitted games, and 3) a questionnaire two months after the lecture during the first semester course "Foundation of CS". At TU Graz, students of CS need to pass the course "Foundations of CS" as part of the orientation period (STEOP). Therefore, students of the course were asked to fill out a questionnaire concerning their experiences and impressions of the last three months. Table 1 provides an overview of the data collection process at the different times.

Table 1: Chronological overview of the data collection process

	MOOC	Offline Lecture	Lecture "Foundation of Computer Science
Time period	05.08. – 30.09.2019 (7 week online course + 1 week offline lecture	18.09. – 25.09.2019	December 2019
Participants	In total 643, finished: 76	In total 102, finished: 76	62
Data collected	Course engagement Final online survey*	ARS feedbackr responses, Pocket Code programs Lecture evaluation**	Post-questionnaire with LimeSurvey***

* This survey is part of every MOOC on iMooX.
**During the lecture, the ARS "feedbackr" (https://www.feedbackr.io/) was used to allow students to ask questions and to give suggestions about the content and organisation of the course in an anonymous way. Feedbackr is a web application, which allows students to access a survey by entering a defined session code, available via (mobile) devices.
***The questionnaire was generated with LimeSurvey (https://www.limesurvey.org/), an open source online survey tool which allows the creation of secure and anonymous surveys.

11. Data Analysis

With the help of questionnaires, we collected quantitative and qualitative data. Quantitative measures were used to evaluate the MOOC itself, while qualitative measures were used to ask open questions about improvements and enhancements, thus exploring the concept of a pre-MOOC more broadly. For evaluation, two kinds of 5-point Likert Scales were used (Sullivan and Artino, 2013). The first asked for agreement/disagreement (1: strongly agree, 2: agree, 3: partly, 3: disagree, and 4: strongly disagree), and the second used a grading system (1: highest to 5: lowest).

For the game analysis, one final game will serve as an example and will be described in more detail. First students had to generate a storyboard to define the genre, theme, and goal of their game (Spieler and Slany, 2018). These game design strategies and decisions were part of the MOOC lecture 6. Second, they created small games which should have fulfilled the following requirements: stick to a concise structure (title, introduction, end screen), use different Mechanics, Dynamics, and Aesthetics (MDA) (min. 3, e.g., points, levels, narrative), and set the level of control. MDAs are defined as key components required for building games (Hunicke et al., 2004). The level of control requires students to integrate different concepts for moving objects such as animations or touch-actions (simple), device sensors or buttons (advanced), or clones or physics bricks (expert).

12. Results and Discussion

Based on the data collection and analysis, the following section presents the main results and findings of the MOOC, the offline lecture and the post questionnaire, shows different engagement levels, and proves that the pre-MOOC "Get Fit in CS" acts as a successful example of a preparatory course for entering the field of CS.

13. MOOC

Participants: At the end of September 2019, a group of 643 participants signed up for the MOOC. In total, 237 participants completed the first lecture, 181 the second, 131 the third, 104 the fourth, 93 the fifth, 81 the sixth, and 76 completed the seventh and last lecture. Hence, 12% of the participants completed the MOOC. The average age of the participants was 27.01 years, while 60.61% were male and 39.39% were female. Most of the participants were from Austria (82.83%), and 14.14% from other German speaking countries. Figure 4 illustrates the answers of the online questionnaire. Participants who have completed all lectures of the MOOC are required to fill out this questionnaire to receive a certificate of participation. Answers are displayed with the average values.

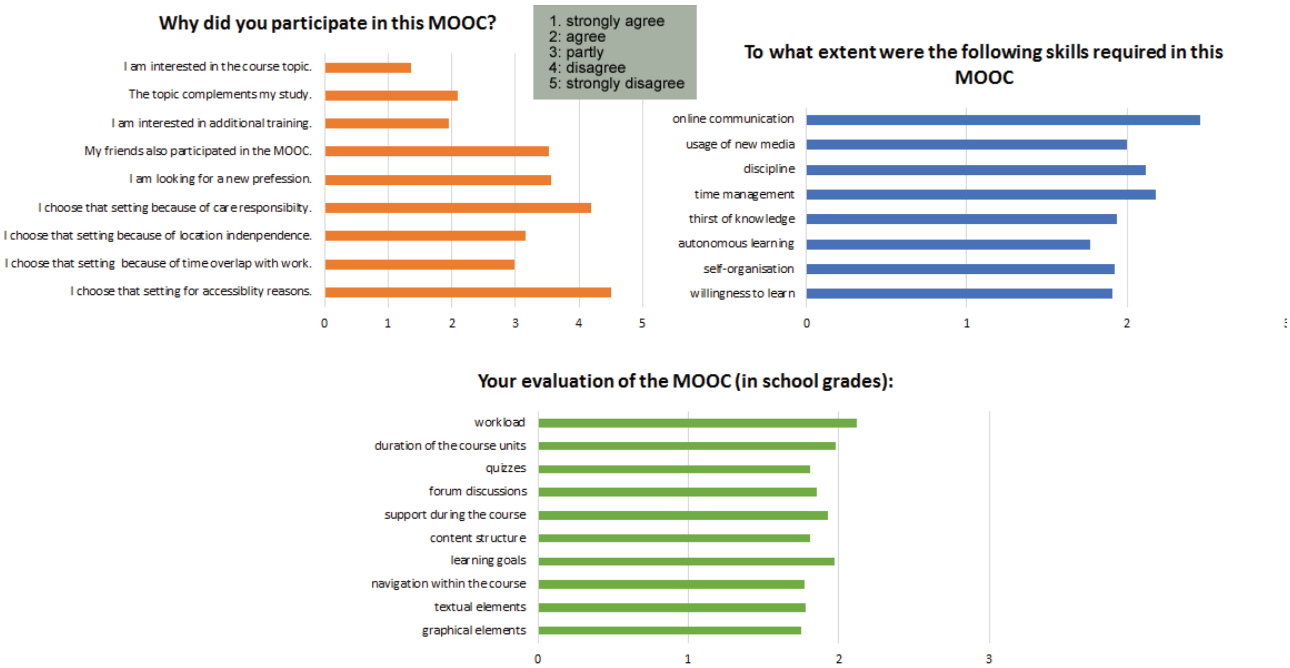

Figure 3: MOOC survey evaluation. a) students' motivational aspects, b) skills required for the MOOC, and c) students' evaluation of the MOOC. Answers are displayed with the average values

Motivation (Figure 4a, in orange): Questions about motivational aspects show mixed results: On the one side, participants strongly agreed or agreed that they are interested in the course topic (1.35) and that the topic complements their study (2.09). This shows a high self-motivation of students to complete the course. On the other side, they strongly disagreed or disagreed that they completed the course because their friends also participated in the MOOC 3.53), that they are looking for a new profession (3.56). Since the majority of the participants were students (67%), these results are not surprising. Examples for open answers were "I want to prepare myself for the study of CS" or "to refresh my skills".

Skills (Figure 4b, in blue): Most agreed answers on skills required by the MOOC were willingness to learn (1.91), self-organisation (1.92), and autonomous learning (1.77). It can be argued that prospective students have already been encouraged to apply skills that are essential for a study.

Content Design and Organization (Figure 4c, in green): The students were asked to grade the MOOC on a scale from 1 to 5 (in school grades). Positive ratings were given for design elements, such as graphical/textual representation (~1.85), and the learning goals, content, and workload (~1.93). Content or navigation received the best rating (~1.73). The overall concept of the MOOC received an average grade of 1.53.

Finally, they rated their engagement with 2.01 on average. 60% stated that they never read or left a post, and 27% used the forum up to five times. Many exercises had to be submitted within a forum, and in total 341 forum entries were made. Some students appreciated the possibility to see code from others and get direct feedback, and some disliked the forum submissions (this was also mentioned in the evaluation of the lecture as well as in the personal feedback given during the lecture). Most of the students dealt with the course topics one to three hours a week (48%) and a percentage of 81% stated that the online course fostered their interest for this topic. Open questions to likes/dislikes showed that participants liked the videos, using the Pocket Code app, Python programming, the examples, quizzes, forum discussions, and the general structure of the MOOC. On the one hand, students mentioned that the Java examples were too difficult for beginners, on the other hand they asked for more examples with textual programming languages like Python instead of using Pocket Code. This is in line with the feedback during the lecture and shows the need to specify the target group of the MOOC more clearly (i.e., beginners). 72% of the participants said that they would like to participate in other MOOCs like this, and 94% stated that they would recommend this MOOC. In conclusion, it can be assumed that for those who completed the MOOC, the concept was very satisfactory.

Discussion: The concept of a pre-MOOC for CS was seen as a welcome offer that was gladly accepted by a certain group of first-year students. Differences in perceived usefulness could be detected between students who already had knowledge in CS and those who were beginners. The MOOC was advertised in newsletters, during registration at the university, at the student service office, and via social media platforms. However, many students already had some or a lot of knowledge in CS and only wanted to gain 1.5 ECTS. These students said that the MOOC was too easy and wished to have more textual programming languages. In the future, the target group (students with no background knowledge in CS) must be communicated more clearly.

Furthermore, the 40% female participation rate in the MOOC provides evidence that gender-sensitive education is a key factor for engagement and collaboration. Female students in particular recognize the absence of female role models in technology with whom they can personally identify (Lockwood, 2006). Female lecturers in MOOCs and at university in CS can break down the stereotyped expectations. The literature suggests that if female students are exposed to strong female role models, they rarely express stereotypical thoughts. Steady contact with successful women (e.g., female professors) leads to higher career ambitions and stronger implicit self-images (Asgari et al., 2010).

14. Offline lecture

Participants: Initially, 102 students signed up for the offline lecture. 76 students actually received a final grade (23 females, 53 males). A total of 59 students were first semester students from different degree programs such as computer science, electrical engineering, software engineering, or psychology.

Audience Response: Students were encouraged to ask and answer questions via the ARS feedbackr and 75 students did so. The students' answers can be categorized in questions regarding the organisation of the course (e.g. "Is there an exam in the end", "Which smartphone is required to use the app Pocket Code"), general comments about the content ("I find it cool that you point out the role of women in the technological progress"), and specific questions regarding the content of the course ("Can you explain the Java program again?", "Can you explain why the symbols i, j, temp, ++ are used in the program?"). The ARS was also used to ask the students about the type of school they came from and their choice of study to better understand the students' motivation and concerns. More than half of the students, who participated in the survey, said that they came from a college/upper secondary school providing a general education. This information led to the assumption that more than half of the students had little to no programming experience.

Lecture Evaluation: At TU Graz students are able to evaluate lectures anonymously and 27 students have used this possibility. Most of them rated the workload as adequate (55.6%) or high (25.9%). The level of difficulty of the course was mostly considered adequate (66.7%) and the majority was satisfied with the course (62.6%). Students enjoyed working independently, designing their own games with Pocket Code, feedbackr, the flipped classroom scenario, the helpful and friendly lecturers, and the opportunity to meet new people before the university classes started. But students saw also room for improvement: the order of the lectures of the MOOC should be changed slightly, a greater emphasis should be put on Python programming basics instead of Pocket Code and Java (in the MOOC as well as in the offline lecture), and attendance should not be mandatory.

Pocket Code Projects: Altogether a total of 113 Pocket Code games were submitted (some students attended the lecture without grading). After the first submission, students received a peer review and a remix from another student and were allowed to revise and resubmit their game to receive extra points. For this paper, a game by a 19-year-old female student is analysed in more detail, see Figure 5. This game has the name "Das große Meeresquiz" ("The big ocean quiz") and is considered an action game with an underwater theme. The goal is to solve different mini-games (e.g.; catch/avoid, shoot). The characters are controlled by the tilt sensor, and physical attributes such as gravity are used for the automotive objects. The game has all screens integrated (title, introduction, end), uses different MDAs (four levels, points, a high score, countdowns, visual and sound feedback). The peer review suggests improvements for navigation (an additional start button) and describes bugs in the high score. For the remix program the first level was extended (more obstacles, different scores). For the bonus task, a narrative storyline was added.

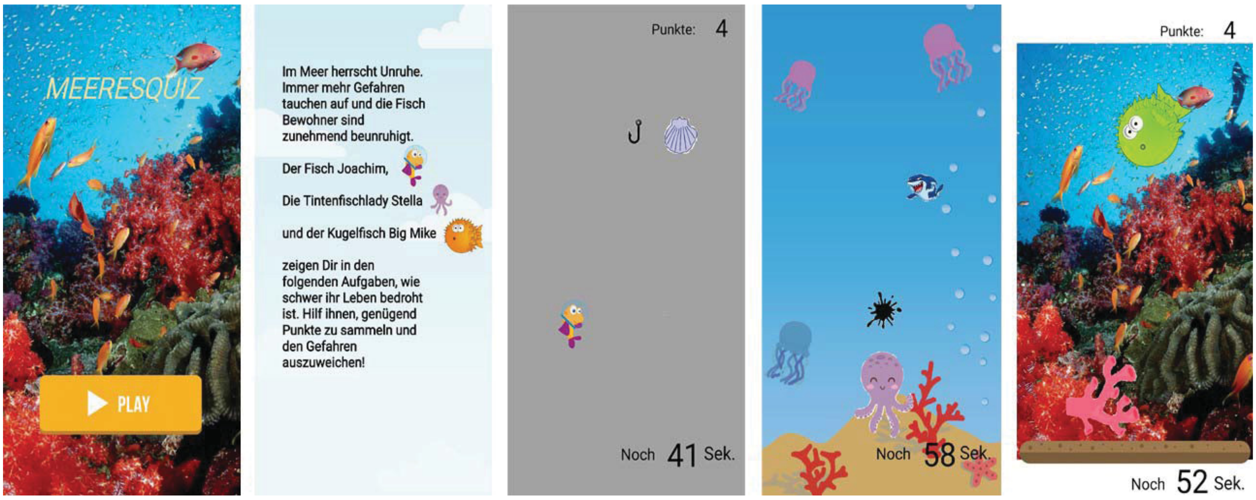

Figure 5: The example game "The big ocean quiz" a) title screen, b) story line, c) catch/avoid items, d) shoot sharks with ink, and e) a pinball game with a blowfish.

Discussion: The results indicate that a safe environment was created in which students were encouraged to actively participate through challenges, activities, questions, and discussion. In general, many students feel uncomfortable asking questions in lectures with big groups (Margolds, Fisher, and Miller, 2014). Particularly in CS degree programs, students have different background knowledge as the integration and intensity of CS topics into high school curricula varies between schools and countries. Years of literature show that female learners in particular tend to be less comfortable than their male colleagues when asking questions, and thus are left with lingering doubts regarding the material (Frieze and Quesenberry, 2015). The MOOC and the ARS-tool allow students to study the content of a lecture at home can build self-confidence in women.

With interactive exercises, the lecturers wanted to foster feelings of enjoyment, which have a positive effect on students' motivation and outcome (Filsecker and Kerres, 2014). Additionally, the peer review and the presentation of the individual Pocket Code projects in front of peers, engages the learners, cultivates their sense of ownership, and gives them the opportunity for sharing different artefacts (i.e., via the Catrobat Community, Papert, 1993).

15. Post-Questionnaire

Participants: The online questionnaire was conducted in December 2019, with students from the lecture "Foundation of CS". A total of 62 students (16 females, 46 males) completed the online questionnaire.

Answers of students, who completed the MOOC: A total of 11 students stated that they completed the MOOC "Get FIT in CS". Of these, five students agreed that the topics that were covered in the MOOC were supportive for the mandatory lectures "Design your Own App" and "Foundations of CS". Three students said they were "not sure" if the MOOC contributed to a better understanding of the lecture(s) and three other students rated the MOOC as not supportive. One of these students argued: "It is better to learn Python and C instead of Java and Pocket Code, because these programming languages have no relevance for CS studies at TU Graz."

Answers of other students: The students, who did not complete or participate (in) the MOOC, were asked, which topics or applications should be discussed in a pre-course. The answers included "basis concepts of programming", "Python for beginners", "more Python, less Scratch and Pocket Code", "introduction to the programming language C", "binary/hexadecimal number system", "sentential logic", "Linux", "Git" (version-control system), or "relevant software for CS students". Except for the programming language C, Linux and specific software solutions, all other topics were covered in the MOOC, respectively. 33 of these students said that they had experienced some lack of knowledge, which related to not having CS lessons at school. Of these, 24 students stated that they had graduated from a college/upper secondary school which provided only a general education or from a vocational school with no technical specialization. This result supports the claim that many (Austrian) schools do not provide an in-depth view of CS.

Discussion: The purpose of the MOOC was to introduce different concepts of programming (loops, conditions, variables, etc.) by using a block-based and a textual programming language. The focus was not on a particular programming language, but to apply these basic programming concepts and to get a feeling for syntactical and semantical issues. Some students were not happy with the blocked-based programming approach and asked for more "relevant" software and programming languages. In a systematic literature review, Noone and Mooney (2018) tried to answer the questions whether there are "any benefits of learning a visual programming language over a traditional text-based language" and whether "the choice of first programming language" makes a difference. Noone and Mooney (2018) conclude, that "the actual choice of what tools to use does not matter, within reason." It is about the conjunction with effective pedagogical and didactic actions. Anyway, Python and Java are characterized as "good" first programming languages. The use of visual programming languages in the first step is announced as "clearly beneficial" as it helps to increase student's learning motivation, interest, and self-efficacy (Weintrop and Wilensky, 2015; Tsai, 2018).

16. Conclusion

In this paper we present the design and evaluation process of the pre-MOOC "Get fit in CS" and the corresponding offline lecture. We put a special emphasis on the students' different background knowledge in CS and female students. The results show that students of various degree programs were attracted by the overall concept (the MOOC and in lecture). This indicates the growing importance of basic CS skills in all fields of studies. In general, students have to understand that it is very likely that a general understanding of the main concepts of CS is required or at least beneficial for their future career paths. Thus, it was important that the MOOC foster students' sense-of belonging by showing a broader picture of CS, the use of a gamified approach, and by providing a low entry point.

The major weaknesses reported by the students were 1) the order of the topics/lectures (in terms of difficulty), 2) organisational aspects in regard to the offline lecture (compulsory attendance, grading), and 3) the frequent use of the visual coding tool Pocket Code instead of textual programming languages. At this point, it can be argued that visual coding concepts may be of "limited benefit to high self-efficacy students with programming background during the initial phase" (Tsai, 2018). In this case, Weintrop and Wilensky (2015) suggest to use a hybrid or dual-modality programming environment to address the needs of learners with a different level of prior experiences in programming.

To sum up, the results led to a better understanding of the way in which students engage with the content of the MOOC. We were able to answer our pre-defined research questions by examining the learners' experiences on a broader level and providing insights into their level of engagement. Furthermore, the evaluation illustrates the need to promote gender-sensitivity and equality in lectures in aiding students' digital inclusion. This was reflected in the positive comments via feedbackr and in the results of the lecture evaluation. However, the analysis also demonstrates the difficulty of motivating students with different background knowledge in programming equally and to design a MOOC for different target groups. Consequently, it is necessary to communicate the target group of the MOOC (=programming beginners) more clearly. It can be argued that such courses are important to create equal conditions right at the beginning of their study and especially in CS where students have different knowledge levels depending on their secondary education. To conclude, this MOOC serves as an applicable example for a preprepartory course for CS and the findings act as a basis for further improvements.

17. Outlook

The MOOC and the lecture will be held again in the winter term of 2020. The results suggest opportunities for improvements. For example, this time it will be mandatory to complete the entire MOOC and its exercises before the offline lecture begins. In this way, all students will start with the same basic knowledge in CS and the lecture can focus more on sparking discussions and offering examples instead of repeating content. The forum has received positive and negative evaluations. However, following the Constructionism theory of Seymour Papert (1993) we will stick to this concept of sharing results. In order to improve organisational aspects, the MOOC must follow the structure of the "Mathe-FIT-MOOC". Here attendance is not compulsory and students do not receive any grades. As a result, in the "Fit in CS"-MOOC students were confused by the different regulations and feared that they would receive a bad grade.

References

Alvardo, 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, pp.27–32.

Asgari, S., Dasgupta, N., and Cote, N. G., 2010. When Does Contact with Successful In-group Members Change Self-Stereotypes? In Social Psychology, 41(3), pp.203–211.

Beyer, S., 2016. Women in CS: Deterrents. Encyclopedia of Computer Science and Technology, (2), CRC Group, P. A. Laplante.

Binder, D., Thaler, B., Unger, M., Ecker, B., Mathä, P., and Zaussinger, S. 2017. [GERMAN] MINT an Öffentlichen Universitäten, Fachhochschulen sowie am Arbeitsmarkt - Eine Bestandsaufnahme. Research Report. Study on behalf of the Federal Ministry of Science, Research and the Economy (BMWFW). Available through: http://irihs.ihs.ac.at/4284/1/2017-ihs-report-binder-mint-universitaeten-fachhochschulen.pdf. [Accessed 16 May 2019].

Brackmann, C. P. et al., 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.), pp.65–72.

Carson, S. and Schmidt, J., 2012. The Massive Open Online Professor Academic Matter, In Journal of higher education, Available through: http://www.academicmatters.ca/2012/05/the-massive-open-online-professor/. [Accessed 03 April 2019].

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), p.58.

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, pp.184–190.

Committee on European Computing Education (CECE), 2017. Informatics Education in Europe: Are We All In The Same Boat? Report by CECE, jointly established by Informatics Europe & ACM Europe. Available through: https://www.informatics-europe.org/component/phocadownload/category/10-reports.html?download=60:cece-report. [Accessed 16 May 2019].

Cuff, E., 2015. The Effect and Importance of Technology in the Research Process. In Journal of Educational Technology Systems. 43(1), pp.75–97.

Dormayr, H., and Winkler, B., 2016. [GERMAN] Befragung Österreichischer LehrabsolventInnen zwei Jahre nach Lehrabschluss. Partial report as part of the ibw study - background analysis on the effectiveness of in-company training place threading (paragraph 19c BAG). Available through: https://www.ibw.at/bibliothek/id/416/. [Accessed 15 April 2019].

Ebner, M., Lorenz, A., Lackner, E., Kopp, M., Kumar, S., Schön, S., and Wittke, A., 2016. How OER enhance MOOCs – A Perspective from German-speaking Europe. In Open Education: from OERs to MOOCs. Jemni, M., Kinshuk, Khribi, M. K. (Eds.). Springer. Lecture Notes in Educational Technology. pp.205-220.

Ebner, M., Khalil, M., Schön, S., Gütl, C., Aschemann, B., Frei, W., and Röthler, D., 2017. How Inverse Blended Learning Can Turn Up Learning with MOOCs? In Proceedings of the International Conference MOOC-MAKER 2017, pp.21-30

Ebner, M. and Schön, S., 2019. Inverse Blended Learning – a didactical concept for MOOCs and it's positive effects on dropout-rates. In The Impact of MOOCs on Distance Education in Malaysia and Beyond. Ally, M., Amin Embi, M., Norman, H. (eds.). Routledge. ISBN 9780367026615.

Ebner, M., Schön, S., Braun, C. (2020). More Than a MOOC—Seven Learning and Teaching Scenarios to Use MOOCs in Higher
Education and Beyond. In: Yu S., Ally M., Tsinakos A. (eds.) Emerging Technologies and Pedagogies in the Curriculum. Bridging Human and Machine: Future Education with Intelligence. pp.75-87.

European Commission, 2016. A new skills agenda for Europe. Working together to strengthen human capital, employability and competitiveness. Available through: http://ec.europa.eu/social/main.jsp?catId=1223. [Accessed 16 May 2019].

European Commission, 2015. Dropout and Completion in Higher Education in Europe. Main Report. Available through:https://supporthere.org/sites/default/files/dropout-completion-he_en.pdf, [Accessed 03 May 2019].

European Statistics Eurostat, 2019. Female students under-represented in ICT. Available through: https://ec.europa.eu/eurostat/web/products-eurostat-news/, [Accessed 03 May 2019].

Filsecker, M. and Kerres, M., 2014. Engagement as a volitional construct: a framework for evidence-based research on educational games. In Simulation & Gaming 45, pp.450–470.
Frieze, C. and Quesenberry, J., 2015. Kicking Butt in Computer Science: Women in Computing at Carnegie Mellon University. Dog Ear Publishing.
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.
Informatics Europe, 2015. Informatics education in Europe: institutions, degrees, students, positions, salaries. Key Data 2009-2014. Available through: http://www.informatics-europe.org/images/documents/informatics-education-europe-data-2009-2014.pdf. [Accessed 05 April 2019].
Khalil, H. and Ebner, M., 2014. MOOCs Completion Rates and Possible Methods to Improve Retention - A Literature Review. In Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2014 Chesapeake, VA: AACE, pp.1236-1244.
Khalil, M. and M. Ebner, 2016. When Learning Analytics Meets MOOCs - a Review on iMooX Case Studies. Innovations for Community Services: 16th International Conference, I4CS 2016, pp.27-29, 2016, Revised Selected Papers. G. Fahrnberger, G. Eichler and C. Erfurth. Cham, Springer International Publishing: pp.3-19.
Kahn, K. (2017) A half-century perspective on Computational Thinking. In technologias, sociedade e conhecimento 4 (1).
Kopp, M. and Ebner, M., 2017. La certificación de los MOOC. Ventajas, desafíos y experiencias prácticas. Revista española de pedagogía. 75 (266), pp.83-100. ISSN 0034-9461.
Price, M. and Lapham, A., 2003. Asynchronous Dialogue in Education: towards an understanding of the nature of interactions. In: WWW.
Li, Y., Zhang, M., Bonk, C. J., and Guo, Y., 2015. Integrating MOOC and Flipped Classroom Practice in a Traditional Undergraduate Course: Students' Experience and Perceptions. In: International Journal of Emerging Technologies in Learning, 6 (6).
Lockwood, P., 2006. Someone like me can be successful: do college students need same-gender role models? In Psychology of Women Quarterly
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 paper of the Carnegie Mellon Project on Gender and Computer Science Available through: https://www.cs.cmu.edu/afs/cs/project/gendergap/www/geekmyth.html.
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), pp.424–437.
Noone, M. and Mooney, A., 2018. Visual and Textual Programming Languages: A Systematic Review of the Literature, Journal for Computer Education.
Pappana, L., 2012. Massive Open Online Courses Are Multiplying at a Rapid Pace. Available through: The New York Times http://www.egymodern.com/2011/07/al-nahar-chaneel.html. [Accessed 04 May 2019].
Papert, S., 1993. The Children's Machine: Rethinking School in the Age of the Computer: Bringing the Computer Revolution to Our Schools. In BasicBooks.
Singer, N., 2019. The Hard Part of Computer Science? Getting Into Class. Available through:https://www.nytimes.com/2019/01/24/technology/computer-science-courses-college.html. [Accessed 04 May 2019].
Spieler, B. and Slany, W., 2018. Game Development-Based Learning Experience: Gender Differences in Game Design. In Proceedings of the 12th European Conference on Games Based Learning. pp.616-625.
Spieler B., Oates-Induchovà, L., and Slany, W., in press. Female Teenagers in Computer Science Education: Understanding Stereotypes, Negative Impacts, and Positive Motivation. Journal of Women and Minorities in Science and Engineering.
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), pp.5–8.
Sullivan, G. M. and Artino, A. R., 2013. Analyzing and Interpreting Data From Likert-Type Scales. In Journal of Gradate Medical Education 5 (4), pp.541–542.
Tsai, C.-Y., 2018. Improving students' understanding of basic programming concepts through visual programming language: The role of self-efficacy. Computers in Human Behavior.
Tumlin, N., 2017. Teacher Configurable Coding Challenges for Block Languages. In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education, pp.783–784.
Veilleux, N., Bates, R., Allendoefer, 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, pp.65–70.
Vos, N., van der Meijden, H., and Denessen, E., 2011. Effects of constructing versus playing an educational game on student motivation and deep learning strategy use. In Computers & Education. 56(1), pp.127–137.
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), pp.82–96.
Weintrop, D., and Wilensky, U., 2015. To block or not to block, that is the question. In: Bers, M.U., Revelle, G. (eds.), Proceedings of the 14th International Conference on Interaction Design and Children IDC '15, pp.199–208.
Wing, J.M., 2006. Computational thinking. Communications of the ACM, 49(3), p.33-35.

Wu, B. and Wang, A.I., 2012. A Guideline for Game Development-Based Learning: A Literature Review. In International Journal of Computer Games Technology. Volume 2012, Article No. 8.

Ya-Ting, C.Y. and Chao-Hsiang, C., 2013. Empowering students through digital game authorship: Enhancing concentration, critical thinking, and academic achievement. In Computers & Education. 68, pp.334–344.

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), pp.283–292.