# The hAPPy-Lab: A Gender-Conscious Way To Learn Coding Basics in an Open Makerspace Setting

Bernadette Spieler[1][0000−0003−2738−019X], Maria Grandl[2][0000−0002−4869−9725], and Vesna Krnjic[2][0000−0001−9555−4556)]

[1] University of Hildesheim, Universitätsplatz 1, 31141 Hildesheim, Germany
bernadette.spieler@uni-hildesheim.de
[2] Graz University of Technology, Rechbauerstr. 12, 8010 Graz, Austria
maria.grandl@tugraz.at
vesna.krnjic@ist.tugraz.at

**Abstract.** In Computer Science, and particularly in the context of Maker Education, students should try out new technologies (including coding) or craft techniques without fear of failure and in a playful way. Studies have shown that learning programming through tinkering appeals to boys more than girls. Taking that into consideration, tools and tasks can make a huge difference in an open learning and teaching environment. These observations are supported by the results of a pop-up-makerspace event for children and teenagers between the ages of 10 and 14. The "MAKER DAYS" for kids took place in the summer of 2019 at Graz University of Technology and attracted 132 children for 4 days. The main goal of the event was to enable authentic learning experiences and to try out new technologies. Participants could choose from a variety of activities, including digital fabrication with 3D-printing, soldering, programmed embroidery, coding, and robotics. Five workshop areas focused on coding skills. The "hAPPy-Lab" acted as a starting point to practise Computational Thinking as well as to learn the basics of coding by developing an app. For example, participants with minor or no coding skills, who wanted to create an embroidery design or use a microcontroller, were asked to visit the hAPPy-Lab first. The hAPPy-Lab implemented a carousel activity and the participants were supported by peer tutors. In this paper, we present the didactic and educational environment of the hAPPy-Lab and suggestions for a similar environment in school.

**Keywords:** Open Learning Spaces · Maker Space · Girls · Creative Coding · Improving Classroom Teaching · Teaching/Learning Strategies.

## 1 Introduction

Making is seen as a promising didactical approach in school to promote important skills such as creativity, collaboration, and problem-solving. However, making is not so much about a fully equipped makerspace, but rather about the development of a maker mindset: Being self-confident and motivated to implement one's ideas [3].

A "maker" is a person who builds, creates, disassembles, extends, redesigns, finds solutions, or implements his or her ideas, which can be anything. There is no clear definition of a Maker's creative area. However, the new thing that has defined the Maker community for some years now, at least since the rise and omnipresence of smartphones and tablets, is the use of modern technologies along with traditional craftsmanship and the related materials (e.g., craft stuff) and tools (e.g., sewing machine). Characteristical maker tools are digital fabrication tools such as 3D printers, laser cutters, and programmable boards [18]. Maker Education creates opportunities to engage children in crafting and tinkering by building creative designs or (digital) projects, thereby learning fundamentals of computer science, electronics, or design and engineering practices, specifically, science, technology, engineering, and mathematics (STEM, or STEAM when art is included) [10].

The MAKER DAYS for kids, which took place in the summer of 2018 and 2019 at Graz University of Technology (TU Graz) in Austria, acted as a playground for the implementation of the basic principles of Maker Education in an open teaching and learning setting for children and teenagers aged 10-14. With the MAKER DAYS, new learning experiences with digital technologies should be enabled, i.e., combining technical educational content with do-it-yourself activities and social aspects. Results of the MAKER DAYS 2018 showed that participants who already had some basic programming skills found it easier to develop innovative product ideas [8]. Three workshop areas (robotics, physical computing, and programmed embroidery) focused on the creation of a program while using a defined development environment [7]. In 2019, the hAPPy-Lab was introduced to act as a starting point to practise Computational Thinking (CT) and to learn the basics of coding by developing small programs with the help of the app Pocket Code.

In this paper, we focus on the learners' experiences in the hAPPy-Lab by asking how to efficiently teach the basics of programming within a short time and with the goal of empowering girls through playful and creative coding activities in an open makerspace setting. Based on the results of the MAKER DAYS 2019, the authors comment on the learning activity design in terms of flexibility, efficiency, gender ratio, popularity and creativity in the context of an open learning and teaching environment.

## 2 Computer Science and Maker Education

There has been a growing interest in teaching students to program to prepare them for the demands of our increasingly digital society [4]. Computational Thinking (CT) and programming are often referred to as the literacy of the $21^{21}$ century [23]. This movement goes well beyond the idea that "we need more software developer" [6]. Consensus is growing that CT and programming are critical skills for all and are quickly becoming a new learning domain, on par with reading and mathematics.

The concept of Makerspaces is more related to creating tinkering-spaces for children, promoting active participation, knowledge sharing, and collaboration among children through open exploration and creative use of technology [14]. Smaller "versions" of this concept could also be integrated into classrooms in the form of maker education [5]. Digital tools, including low-cost microcontroller platforms, visual coding applications, or online community infrastructures can bring the idea of making and tinkering to classrooms [10] and create opportunities to learn about programming principles.

The most difficult issue of making is to define and implement one's projects. Identifying a problem or finding an idea for a (digital) project are important qualifications. However, the possibility to choose one's project increases the identification and consequently, the motivation to work on this project [17]. Many things are demanded from a teacher in an open learning environment as they need to accompany and supervise the individual implementation of their students' ideas. The teacher has to accept that he or she might not have an answer to a question immediately. In the best case, the teacher has to structure a learning activity in a way that it is open, but has enough instructions, so that nobody is overwhelmed. Asking students to act as facilitators or experts and to help the other students can take some of the pressure off the teacher, but forces the teacher to (probably) change the understanding of one's role. From a student's point of view, learning is different in an open makerspace setting, compared to the content and methodology of a regular class in school. They are allowed to copy and share their ideas and to define and solve their problems. Sometimes such activities are focused on designing games or supporting students to build playable artefacts [12]. In this case, strategies that focus more on design and creativity a game and less othe programming itself can provide a promising way to attract all and female teenagers in particular for coding activities [20]. With digital designs, activities coded to be more "male" such as engineering and CS can be combined with activities that are considered as traditionally "female", such as crafting and sewing. For example, e-textiles have a huge potential to attract all genders. Decorating cloth, sewing circuit designs on clothes, for example with conductive thread, or programmable embroidery machines are often part of maker spaces. As a result, young people have something to wear that they could show to others.

Another crtical aspect of maker education is creation of safe learning environments in a more gender-sensitive way, e.g., taking into account that young people with different levels of prior knowledge in CS visit such courses. Therefore, mentoring or tutoring programs are key elements to introduce especially girls to technical subjects and to awaken their interest for CS [22, 1]. Female tutors with whom they can personally identify are described as the most effective. Furthermore, the research of Krieger, Allan, and Rawn [9] observed tinkering strategies across genders in undergraduate students of CS via interviews and a questionnaire. According to the authors, tinkering means exploring and is generally considered as an informal practice. Thereby, they see it on the same level as using problem-solving abilities or students asking for help. Results showed different definitions or perceptions of tinkering activities by gender, and that girls are less likely to see themselves as tinkers. Thus, the authors proposed to think of teaching tinkering for non-tinkerers as well. This is also consistent with the findings of Beckwith et al. [2] who stated that male students seem to benefit more from tinkering activities. However, tinkering can help girls to gain valuable information about the features and increase their self-efficiency. Low tinkering interactions and low self-efficiency occur in girls if they use environments that are described as too complex. The study concludes that gender differences exist in the way students solve problems, which may indicate a need for supportive feature designs.

## 3 The Event: MAKER DAYS for kids

The MAKER DAYS for kids are an educational event for children and young people aged 10 -14 year, with the overall goal to help children and teenagers become more fluent and expressive with new innovative technologies as well as traditional tools and materials [7]. Inspired by the concept and results of the first MAKER DAYS in Bad Reichenhall (Germany) in 2015, the MAKER DAYS took place at TU Graz in the summer of 2018 and 2019. As part of the university's summer course program for children and teenagers, the MAKER DAYS offered different activities in an open makerspace setting, where more than 50 participants per day were supported by (peer) tutors. In 2019, the pop-up makerspace was open for four days and participation was limited to two days. Registration was required to ensure a balanced gender ratio. In 2019, 58 children and teenagers (45% female, 55% male) participated at the MAKER DAYS on the first day, 57 (44% female, 56% male) on the second day, 65 (46% female, 54% male) on the third day and 59 participants (47% female, 53% male) on the last day.

Four lecture rooms with an overall size of 400 $m^2$ were transformed in a temporary makerspace with separated workshop areas, focusing mainly on digital fabrication (creating files for the 3D-printer, vinyl cutter, laser cutter, sticker machine and embroidery machine), coding and robotics (physical computing with programmable boards, e.g., BBC micro:bit, Calliope mini, creating projects in Scratch, choose from selected coding tutorials, app design, and coded embroidery with Pocket Code, solving tasks with the Thymio robot, the mBot, and the Ozobot), electrical engineering (soldering, building electric circuits, projects with littleBits), crafts and arts.

From a participant's point of view, the day started with a guided makerspace tour. Then, participants could choose from different activities and work on their projects in the corresponding workshop areas. In 2019, every participant got an empty sticker card, which also acted as a name tag. The participants could collect stickers if they spend a certain amount of time on a specific activity or create a (valuable) product [8]. Each workshop area had a characteristic sticker and additional star stickers for outstanding projects.

## 4 The hAPPy-Lab

The idea of the hAPPy-Lab was to teach children the basic principles of programming (e.g., loops, conditions, variables, communication between objects) and to practise CT (e.g., writing an algorithm, debugging, decomposition). The results of the MAKER DAYS 2018 showed that basic programming skills were required in three main workshop areas or making activities. The tutors argued that they often had to explain the basics in a short space of time to keep the participants' motivation and interest on a higher level. The participants' focus was often more on the final product (e.g., a traffic light, using the BBC micro:bit, 3 LEDs, crocodile clips and cardboard) and not on the iterative (learning) process (e.g., How can I turn an LED on/off with the BBC micro:bit) that leads to the final result. This is why the project head decided to implement a coding activity, where a large number of participants could participate at the same time, independent of their programming experience and learning rate. As well as that, participants should be motivated to create their app (e.g., game, story, animation,etc.) with the Pocket Code app in the end.

### 4.1 The Pocket Code App

For creating games during the hAPPy-Lab we used our coding app Pocket Code. Pocket Code uses visual block-oriented programming language very similar to the one within the Scratch environment. With over 50 million shared projects (as of July, 2020), Scratch is one of the best-known visual programming languages for children and teenagers in the world. According to Seymour Papert [13], founder of constructionism and developer of LOGO, a programming language needs to be characterized by the following two characteristics: On the one hand, it should enable an easy and intuitive introduction to programming ("low floor") [16]. On the other hand, it should allow users to implement complex and sophisticated projects ("high ceiling" and "wide walls")(ibid.) Scratch and Pocket Code, both fulfill those requirements. Accordingly, a programming language should enable the implementation of many different projects to meet the user's different interests and learning methods (ibid., [15]). Programs created by the users show that Scratch appeals to different target groups since the context of a program is not limited to areas related to CS. However, currently Scratch is only available on desktop computers. Considering the enormous growth and low prices in the market of smartphones in the past 15 years we can assume that smartphones will probably be used more by students in future than PCs or laptops. The free mobile app Pocket Code developed by the non-profit Catrobat project (https://catrobat.org) at Graz University of Technology, was one of the first development environment for Android based systems, implementing a block-based and object-oriented programming language according to model of Scratch. In addition, users of Pocket Code can use all built-in sensors (inclination sensor, GPS, face detection, etc.) of their device to control an object.

Furthermore, Pocket Code has a number of extensions that allow users to use external hardware within their created projects. Categories for special hardware such as Raspberry PIs, Arduino, Lego Robotic, or Embroidery are hidden per default and can be enabled in the preferences. The Embroidery extension was especially developed to make the app more attractive for young women from 13 years on. During the MAKER DAYS 2019 participants were able to program self-created patterns and designs that were embroidered on T-shirts, bags, or other fabrics.

### 4.2 Tutors

The results of the first MAKER DAYS event in Bad Reichenhall in the year 2015 have shown that "girls chose more workshops offered by female tutors than boys" [19]. Consequently, a female role model (tutor or teacher) "seems to be a strong supporter to help girls to get in touch with technology." (ibid.) A European-wide study by Microsoft [11], in which 11,500 young women between 11 and 30 were interviewed, showed that the lack of female role models is currently among the top 5 reasons why girls/women are underrepresented in technical professions and studies. Because of this, we decided to ask female students, who have applied for a holiday internship at TU Graz, to support younger kids in the hAPPy-Lab. The students were between 16 and 19 years old and had already worked several weeks with Catrobat/Pocket Code at the Institute of Software Technology. They were also required to participate in the development process of the learning and teaching materials that were used in the hAPPY-Lab.

### 4.3 Activity Design

The hAPPy-Lab implemented a carousel learning activity that consisted of five, more or less mandatory, tasks and two optional tasks. In order to receive a sticker for the hAPPy-Lab, participants of the MAKER DAYS had to solve all five tasks/units. On each table, they could find the corresponding learning materials. The units build upon each other. In the first unit (at the first table), a new "empty" app (project) is created in Pocket Code and the term object was introduced. In each unit, participants are asked to add a new functionality to the app (Unit 1: new project/objects, Unit 2: animation/loops, Unit 3: interaction/messages, Unit 4: collision/conditions, Unit 5: game design/variables). A guided process was defined: There is a task, which should be implemented by the students as independently as possible. New commands and programming concepts are explained on the correlating aid or cheat sheet. To better understand the (logical) program flow and to get to know new commands even faster, the commands (bricks), that are necessary to solve the task/unit, are provided as laminated printouts. In this way, participants had the opportunity to place the sequence of commands (for a specific object) on the table and discuss the program structure with other students and the tutor. After this unplugged coding activity, participants are asked to solve the task with Pocket Code. If all tasks were completed by the participants, he or she could do an additional task (Cookie-Clicker app) or start with the development of an own app. The whole lab is illustrated in Fig. 1.


**Fig. 1.** The hAPPy-Lab: A carousel learning activity with five learning units and tasks to introduce participants to the basic concepts of coding by using the app Pocket Code.

Another optional activity was offered in the lab (see Fig. 1 left): Children who were waiting for a free place in the carousel coding activity or those who just wanted to try something out for a short time could tinker by using special robots called "bigtrack Rover". To control the wheels, these robots do not require an additional connection, e.g., via Bluetooth, but it is controlled with four light sensors. Children used a pre-installed program and they only had to call the predefined messages within the code to activate the different light sensors. A tutorial of the required bricks was part of the printed-out mazes, so they could try this activity without further guidance and thus had a first introduction to the app and coding.

## 5 Findings

The findings of the hAPPy-Lab elucidate the setting of the lab and explore the activities more broadly by providing insights into the different learning units and workshop days. Throughout the week, participant observation notes were recorded during and immediately after each day. Based on those observations, short-term adjustments on the activities were made. Furthermore, we tracked if participants who visited the lab intending to gain programming experiences visited other stations, for example, the programmed embroidery workshop where participants had to use the Pocket Code app again.

### 5.1 The hAPPy-Lab in Numbers

A total of 76 individuals (42 girls, 34 boys) visited the hAPPy-Lab during all workshop days (Monday: 25 participants, Tuesday: 11 participants, Thursday: 19 participants, Friday: 21 participants). Some of the children visited the Maker Days for two days (either Monday and Tuesday or Thursday and Friday). Since the hAPPy-Lab was promoted as a pre-activity for other workshop areas focused on coding skills, more participants decided to visit the hAPPy-Lab on their first day. The numbers of participants for each activity within the hAPPy-Lab (see Section 3.2) are summarized in Tab. 1.

**Table 1.** Completed activities of hAPPy-Lab participants.

| | **Individual visits** | **All units** | **Cookie Clicker Challenge** | **Own app** | **Rover activity** |
|---|---|---|---|---|---|
| **Female** | 42 | 32 | 19 | 1 | 5 |
| **Male** | 34 | 28 | 13 | 3 | 6 |
| **Total** | **76** | **60** | **32** | **4** | **11** |

60 participants, this is 45% of all participants, finished all units. Almost all participants experimented with the rovers while waiting for the "all units"-activity. 11 participants only tried out the Rover without any further engagement in the hAPPY-Lab. 32 participants (19 girls, 13 boys) finished all units and the final Cookie Clicker challenge. One girl and three boys finished all units and created their own app afterward. Five students (4 female, 1 male) visited the lab on two different days to work further on the units.

3 students (2 male, 1 female) received a star sticker for their completion. On average, participants needed 105.26 minutes to finish all five units. The fastest with all units was a boy with the minimum time of 31 minutes. The longest time needed to complete all units was 325 minutes by a girl (almost 6 hours). On average participants spent 91.26 minutes within the hAPPy-Lab working on different activities. Girls spent on average 87.68 minutes within the hAPPy-Lab and boys spent on average 94.41 minutes. 5 participants visited the lab two times (4 girls, 1 boy).

Additionally, we examined if participants who programmed their embroidery designs in Pocket Code in the textile studio have visited the hAPPy-Lab beforehand. 26 children created their embroidery designs in the textile studio and 7 children (2 male, 5 female) completed all units in the hAPPy-Lab before the embroidery activity, 5 students participated in the hAPPy-Lab afterwards, 2 girls only did the rover activity beforehand and 12 participants (6 male, 6 female) did not visit the hAPPy-Lab at all.

### 5.2 Observation Notes

The whole lab was designed with a focus on gender-sensitive issues by having female tutors and a progressive course introducing the foundations of programming. During the guided makerspace tour, first, the hAPPy-Lab was promoted as "If you have no prior knowledge in coding, you better start here." The team had the feeling that especially the girls took this advice very seriously. Consequently, on the first and third days, the lab had been fully exploited and many participants were waiting for the next slot. Furthermore, the team recognized that many participants wanted to complete all units within a short time to be able to attend other workshops where those basics of programming were needed.

The idea was that each of the four tutors introduce one unit and supervise 6 children. After completing one unit, the children go to the next table to work on the subsequent unit with a different tutor. For most of the participants it was their very first introduction to coding thus they needed more time for unit 1 and unit 2 to get familiar with the tool and the basic concepts/vocabulary such as algorithms and loops. So when the first six participants started with Unit 1, not only the other children had to wait until the tutor from Unit 1 was available again, but also three of the tutors had nothing to do. The whole material (instructions, bricks, activities etc.) was glued to the table. That made it impossible to do a unit at another table and also the room was very small so that it was not possible to rearrange the tables. This was the reason why we decided that the tutors escort one group of children through all five units and move with them from table to table. On the one hand, this allowed each group to work at their own pace and not feel stressed as the next group was already waiting. On the other hand, fewer children could visit the lab at the same time. Starting with the second day, it was also said that more time should be scheduled to get the sticker for the hAPPy-Lab.

Children who were waiting were asked to come back at a later time or six children could try out the Rover activity at the same time, which was intended as an introductory activity. In contrast, most of the other workshops had a fixed start and end times. In general, less than one hour was planned for all units. Children who completed all units, and the corresponding challenge or an own app needed much more time (up to 2 hours). In the beginning we planned that only those who completed all units and the challenge would get a sticker in their pass. From day two onwards a sticker was given as soon as all units were completed.

With the adjustments of day one, the children were much more relaxed and wanted to try out several activities of the lab. Some just visited to try out the Rover activity. This could be completed in about 10 minutes. Three of the children also spent a longer amount of time to create their own gaming apps by making extensions of the Cookie Clicker challenge or they used the tutorial cards with whole games explained (for example the one girl who spent almost 6 hours in the lab).

## 6 Discussion

The hAPPy-Lab has two central findings: on the one hand, results lend evidence to the assumption that coding workshops that were promoted as "learning the basic knowledge" interest predominantly female participants; on the other hand, it shows that those who visited the hAPPy-Lab mostly finished all units. 74.07% who entered the hAPPy-Lab also received a sticker for completing all units. This let us conclude that the overall structure of the lab (guided process, step-to-step approach, tutors) strongly encourages participants to finish the activity, i.e., to have a finished app/game at the end. As well as that, tutors of other workshop areas agreed, in an qualitative interview after the event, that participants performed better in the coding activities after finishing the hAPPy-Lab activities.

The hAPPy-Lab was mainly attended by girls. Compared with the other workshops of the Maker Days only two more workshops had more participating girls than boys: the programmable embroidery workshop with 18 girls and 8 boys and the Smart-Lab (physical computing with programmable boards) with 17 girls and 13 boys. In contrast, the Code Garden (e.g., coding tutorials, Scratch, textual programming languages, etc.), where basic knowledge in programming was of advantage, only 18 girls took part but 42 boys.

The observation notes of the hAPPy-Lab activities during the Maker Days let conclude that the setting of the lab and the carousel learning activity in particular (with different times required for each unit) was not ideal for this open learning space environment. The room offered space for 24 people. Due to the carousel learning activity and the necessary change between the tables, far fewer than planned were able to attend at one time. Since especially in the morning everyone came to the lab at the same time, a fixed seat with one tutor per table for all units and without the formation of group would have been a better solution.

## 7 Conclusion & Outlook

To current study showed an effective way how 11-14 year old children can learn the basics of coding in an open makerspace setting and whether this lab also appeals to girls in particular, e.g. through the support of female peer tutors and an easy introduction to coding with the visual coding tool Pocket Code. The hAPPy-Lab provides evidence for the learning of coding in a short time, but also highlights some difficulties. For example, children must have the possibility to work at one's own pace. Some felt stressed by working as a group in such a carousel learning activity. Those who were faster wanted to go already to the next station whereas some needed more time. Therefore, it is difficult to set a fixed time frame for the hAPPy-Lab. Especially in children's first programming attempts it is essential to have time and to get support especially for girls [21]. If important concepts are taught too fast and are not understood, the chance that those visit another coding activity at the MAKER DAYS will probably be smaller.

All the documents that were used in the hAPPy-Lab are published as Open Educational Resource (OER) on the project website (www.tugraz.at/go/infogrubi). Teachers are allowed to copy, edit, and redistribute the learning material. This way, the carousel learning activity can also be implemented in a classroom setting with a teacher and 2-4 students as tutors or experts. Students need to bring their own device (BYOD) and have to download and install the app Pocket Code (available for Android and iOS). As students can create their own embroidery designs with Pocket Code, CS teachers and handicrafts (textile and technical design) teachers can work together to exploit the full potential of Pocket Code and to meet the students different interests.

The setup and operation of a makerspace for schools are often connoted with considerable effort and require higher financial resources. However, instead of buying special, expensive manufacturing equipment, schools and teachers should think of how they can help their students to develop a Maker-Mindset in the context of a rapidly changing digital society and should look at the possibilities given on site.

In August 2020, the MAKER DAYS for kids will take place again at TU Graz. Due to the Covid-19-regulations, the project head applied some adaptations to the concept. The number of participants was limited to a maximum of 36 participants per day. This year, there will be a small version of the hAPPy-Lab where participants can apply for a basic course (app design) or submit their own project ideas, which are approved by peers and tutors.

## References


1. Archard, N.: Rethinking the problem of gender and IT schooling: discourses in literature. In: Mentoring & Tutoring: Partnership in Learning. vol. 20, no. 4, pp. 451–472. (2012)
2. Beckwith, L., Burnett, M., Grigorenu, V., Wiedenbeck, S.: Gender HCI: What about the software? In: IEEE Computer Society Press. 39, pp. 97-101.(2006)
3. Clapp, E., Ross, J., O. Ryan, J., Tishman, S.: Maker-Centered Learning: Empowering Young People to Shape Their Worlds. Jossey-Bass, San Franzisco. (2016)
4. Committee on European Computing Education (CECE): Informatics Education in Europe: Are We All In The Same Boat? Report by The Committee on European Computing Education. Jointly established by Informatics Europe & ACM Europe, https://www.informatics-europe.org/component/phocadownload/category/10-reports.html?download=60:cece-report. Last accessed 16 May 2020
5. Davidson, A.-L., Price, D.W.: Does Your School Have the Maker Fever? An Experiential Learning Approach to Developing Maker Competencies. In: LEARNing Landscapes. Teaching With Technology: Pedagogical Possibilities and Practicalities, vol. 11, pp. 103-120. (2017)
6. DeNisco Rayome, A. (2016) CIO Jury: 83% of CIOs struggle to find tech talent. https://www.techrepublic.com/article/cio-jury-83-of-cios-struggle-to-find-tech-talent Last accessed 11 September 2020
7. Grandl M., Ebner M. and Strasser A. (2019) Setup of a Temporary Makerspace for Children at University: MAKER DAYS for Kids 2018. In: Merdan M., Lepuschitz W., Koppensteiner G., Balogh R., Obdrlek D. (eds) Robotics in Education. RiE 2019. Advances in Intelligent Systems and Computing, vol. 1023. Springer, Cham

8. Grandl, M., Ebner M., Schn, S., Brnner, B.: MAKER DAYS for kids: Learnings from a Pop-up Makerspace, Submission for the 11th International Conference on Robotics in Education. (2020, in production)
9. Krieger, S., Rown, A.: Are Females Disinclined to Tinker in Computer Science? In: 46th ACM Tech. Symp. on Computer Science Education, pp. 102-107. (2015)
10. Martin, L.: The Promise of the Maker Movement for Education. In: Journal of Pre-College Engineering Education Research (J-PEER), vol. 5, no. 1, art. 4, (2015)
11. Microsoft: Microsoft-Studie: Kreativität ist Schlssel fr mehr Frauen in MINT-Berufen [Microsoft study: Creativity is the key to get more women in STEM-jobs] https://news.microsoft.com/de-de/microsoft-studie-mehr-frauen-mint-berufen/ Last accessed 22 Juni 2020
12. Ochsner, A.: Lessons Learned With Girls, Games, and Design. In: 3rd Conference on GenderIT (GenderIT 15), pp. 24-31. (2015)
13. Papert, S.: Mindstorms. Children, Computer, and Powerful Ideas. In: Basic Books Inc, (1985)
14. Rosa, P., Ferretti, F., Guimares, ., Panella, F., Wanner, M: Overview of the Maker Movement in the European Union. (2017)
15. Repenning, A.: Moving Beyond Syntax: Lessons from 20 Years of Blocks Programing in AgentSheets. In: Journal of Vis. Lang. and Sentient Systems. pp. 68–91 (2017)
16. Resnick, M., Silverman, B., Kafai, Y., Maloney, J., Monroy, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J.: Scratch: Programming for All. In: Commun. ACM (2007).
17. Resnick, M., Robinson, K.: Lifelong Kindergarten-Cultivating creativity through projects, passion, peers, and play. In: The MIT Press Cambridge, Massachusetts. (2017)
18. Schön, S., Ebner, M., Kumar, S.: The Maker Movement. Implications of new digital gadgets, fabrication tools and spaces for creative learning and teaching. In: eLearning Papers, vol. 39, pp. 14–25, (2014)
19. Schön, S., Rosenova, M., Ebner, M., Grandl, M.: How to Support Girls' Participation at Projects in Makerspace Settings. Overview on Current Recommendations. In: Moro, M., Alimisis, D., Iocchi, L. (eds.) Educational Robotics in the Context of the Maker Movement, vol. 946. Advances in Intelligent Systems and Computing, pp. 193-196. Springer International Publishing, (2018)
20. Shaer, O., Westendorf, L.,Knouf, N. A., Pederson, C.: Understanding Gaming Perceptions and Experiences in a Womens College Community, In: CHI Conference on Human Factors in Computing Systems, pp. 1544–1557, (2017)
21. Spieler, B., Mikats, J. , Valentin, S., Oates-Indruchova, L., Slany, W.: RemoteMentor: Evaluation of Interactions between Teenage Girls, Remote Tutors, and Coding Activities in School Lessons. In: 7th International Conference on Learning and Collaboration Technologies (LCT). pp. 547–567, (2020)
22. Stoeger, H., Duan, X., Schirner, S., Greindl, T., Ziegler, A.: The effectiveness of a one-year online mentoring program for girls in STEM. In: Computers & Education, vol. 69, p. 408–418. Elsevier Ltd. (2013)
23. Wing, J.: Computational thinking. In: Communications of the ACM, pp. 33-35.(2006)