

sCool – Game-Based Learning in Computer Science Class

A Case Study in Secondary Education

<https://doi.org/10.3991/ijep.v9i2.9942>

Alexander Steinmaurer (✉), Johanna Pirker, Christian Gütl
Graz University of Technology, Graz, Austria
`a.steinmaurer@student.tugraz.at`

Abstract—In this paper we present a case study of the mobile learning game sCool [1]. Based on previous work presented Steinmaurer et. al. in [2] we have expanded our study with the introduction of a second experiment and with new additional aspects. sCool is a multi-platform game that is intended to encourage and support children learning computational thinking and coding in Python. The learning content is highly adaptable; educators can thus create own courses on an individual basis for the needs of their students. These courses involve a concept-learning and a practical mode. First, the students learn a specific concept and in a second step, they have to apply it in a practical task. For this purpose, we created a course to teach some basic programming concepts. Two student groups of different school types participated in class as a formal learning activity. In this paper we present the results of the evaluation of sCool in coding classes. Therefore, we focus on the performance, game engagement, emotions and the perception of the girls. Within this study we found out, that the students are interested in learning to code but do have problems to transfer the learned content to similar fields. We also found out, that there are slightly differences in the performance of the different types of students in terms of gender and school type.

Keywords—Game-based learning, computational thinking, STEM, sCool

1 Introduction

Computational thinking is becoming increasingly important in society today. This is not only a skill related to computer science. Computational thinking enables people to recognize problems and solve them in efficient ways. It involves algorithmic thinking, evaluation, decomposition, abstraction and generalisation [3]. In many countries computational thinking is already part of the K-12 curriculum. It is not constrained to the usage of a computer and so certain skills (for example pattern recognition) can even be taught in the kindergarten [4].

In this paper, we intend to present our experience with the game-based learning tool sCool and its usage in school classes. sCool [1] has been designed and developed in a research collaboration between Graz University of Technology, Austria, and Westminster University, UK. The paper is an extension to a conference paper [2]

extended by additional research findings. In the game school, students learn about computational thinking in an explorative way. The course helps them to solve different problems in an algorithmic way. In this context we wanted to explore whether students can learn different concepts in computational thinking and in coding in an analogous manner with the aim of achieving mobile learning. In addition to this focus, we also tried to consider if the learned concepts can be applied to other problems in the same domain. In order to obtain a deeper understanding of the learning process we were also interested in evaluating the engagement of the players and their emotions concerning the video game. Another relevant aspect was to find out if there are differences in games perception in girls and boys.

The main contributions of this paper can be summarized as follows:

- Introducing the tool sCool in two learning situations in secondary school
- Expanding the evaluation of [2] by a second experiment in another school type
- Evaluating both school types to find out their pros and cons in a learning situation and to see whether students are engaged and motivated by mobile learning
- Discussing the findings with the aim of bringing about an improvement in the pedagogical concept

The paper is organized as follows: First, we give an overview of the status quo in mobile- and game-based learning. We instance different approaches and possible problems that can arise. We then introduce the mobile learning tool sCool and its predefined course including the learning contents. In the course of this we present the game's structure and conceptual architecture. In the following chapter, both experiments in secondary school are addressed. This includes the setup of the experiment, the evaluation instruments, the student's tasks, the evaluation methods and the results. In the results we oppose and compare different aspects such as the school types and the player gender. In the next chapter we summarize our findings and discuss them in accordance with the evaluation results for both experiments. We also give a prospect about the next steps in the further development of sCool.

2 Related Work

Video games are not only perceived as entertainment tools, and pedagogical applications have now been recognized for decades. With the aid of video games different subjects can be learned. Thereby, it is important that the right balance is achieved between covering the learning concepts and the game play [5]. A major benefit in m-learning is, that it does not depend on a specific place or time [6]. The knowledge is accessible anywhere and anytime. When students work independently [7], they learn new concepts that have to be presented in a clear way and afterwards they do self-examination using a test system. M-learning is also highly learner-centred, so the user plays a vital role. The learners can play in a safe environment, where making mistakes is allowed (and desired) [2]. A significant factor for successful learning is the learners' motivation. The engagement correlates with flow of the player. If the tasks are too easy or too difficult, they will be frustrated. A balance between the skill and chal-

allenge levels is needed [8]. Keeping the players engaged is relatively challenging. An immediate feedback from the tasks and interactive challenges can increase this balancing [9, 10]. Mobile learning improves the problem-solving skills of the players in a goal-oriented way [11]. They are encouraged to solve a given problem in any possible way. The focus is not on the implementation it is solving the given problem.

According to Yassine et. al. (2018) [10] mobile-learning applications can be classified in three different groups: learning a whole programming language, learning a special concept and learning fundamental programming concepts (not related to certain programming language). For each category, there are many different tools that pursue different approaches. Beside these programming concepts game-based learning enables other important 21st century learning skills like collaboration, creativity, communication and critical thinking (4Cs) [12].

Since many technical concepts – especially in programming – are on a high-level of abstraction, beginners often become frustrated in the efforts they make at understanding them [13]. Game-based learning makes it possible to simplify those concepts and adapt them to the needs of the learners [10]. According to Lin (2016) [14] there are two causes that make learning to code a difficult task: a lack of problem-solving skills and poor feedback. For beginners it is often hard to understand certain structures and concepts such as divide and conquer. For a starter capturing the whole problem and applying already learned concepts for new or even similar problems can also be difficult. The focus is often teaching the programming language itself (for example the syntax) and not the ability of problem solving. The language is not seen as tool but rather as the learning content. If students get less feedback from the teacher (as is mostly the case in a traditional class) they quickly become frustrated. They do need an immediate feedback to stay motivated [13].

There are many different tools that follow a game-based learning approach in introduce programming. CodeMonkey [15] is a web-based tool that helps educators to teach programming. It is using the programming language CoffeeScript [16] due to its user-friendly syntax and its similarity to natural speech. This enables the solving of problems without thinking about complex syntax and so the problem can be focused [13]. The pupils learn computational thinking by helping a monkey to gather bananas. They learn about statements, interacting with objects, arguments, loops, arrays, functions, conditions, logic, etc. It provides a tutoring system if there are any obscurities or problems. CodeMonkey also provides a teacher's dashboard to track the students' progress and the teachers can see the written code. There is also a pre-designed curriculum for teachers using CodeMonkey in class. Another tool is Lightbot [17], this is a multi-platform (web, iOS, Android) game for beginners. Using Lightbot players solve different puzzles by controlling a robot. The square fields have to be lit by using different coding concepts like statements, procedures and loops. The players do not have to write a code, Lightbot is fully block-based, so it does not depend on a certain programming language.

Based on previous experiences and findings from literature, we aimed at developing a mobile-learning game [1], that is highly adaptive and supports both, pupils and educators in learning [2].

3 STEM Education With sCool

The educational game sCool [1] is a project initiated in a collaboration between Graz University of Technology and Westminster University. It is a pedagogical tool to motivate students in the age of 10-20 for STEM fields. The educational game is designed for mobile applications and consists of two different parts (see also Figure 1): The Unity-based game is currently available for Android devices and a Web platform where educators can adapt the content of the courses. sCool is highly adaptive, and it can be used in different fields of STEM education. The courses in the latest version have a special focus on computational thinking and coding in Python. Thereby the students control a robot that has a crash landed on a distant planet and help it to repair the spaceship and escape from the planet. During this mission they learn different concepts in coding and computational thinking and they also have to apply them to rescue the robot. All maps are newly generated each time depending on the predefined parameters of the web application. The learning content is loaded via a JSON API from the server. All game-related data (quiz answers, game metrics, scores, etc.) are also synchronized with the server. As part of the web application, educators can analyse the provided data to assess the players learning level.

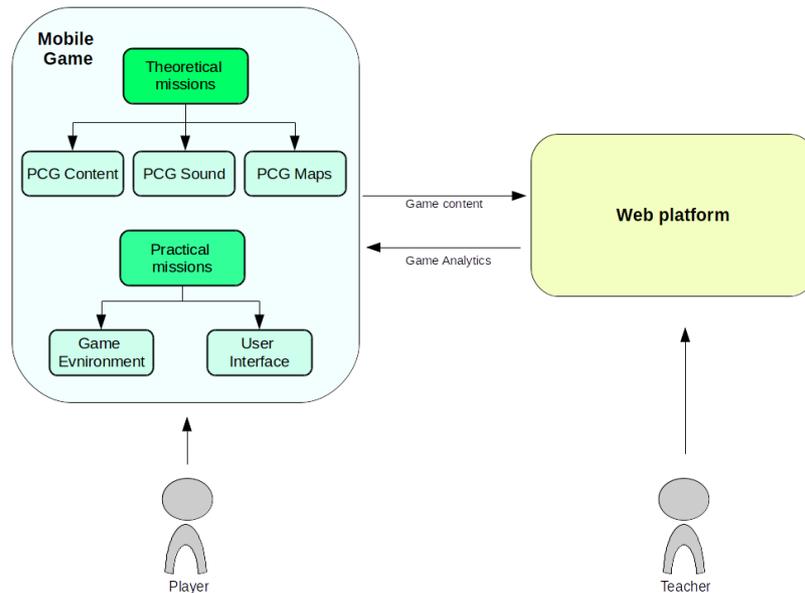


Fig. 1. sCool – Conceptual Architecture adapted after [2].

3.1 Mobile application

The game was first described in Kojic et. al. (2018) [1]. It consists of two different gaming modes, that both pursue different objectives in the process of learning new concepts. Both modes are connected through an underlying story. The players re-

ceived an emergency message from a space exploration team that has crashed on a distant planet. With the aid of a robot, the students should go on a rescue mission and collect all the lost disks, that are necessary to run the space shuttle again. Since some disks are damaged, the players have to complete the code on the disks and test them. In the exploration mode, they learn about different programming concepts that will help them to solve the coding examples.

After each task, the players gain experience and for each achievement, they earn coins. The coins can be used to customize and extend the own avatar. It is possible to buy different items, to make the robot stronger and more difficult levels can be challenged.

Explorative or concept-learning mode: In this mode, the students are charged with the mission of discovering a hostile planet. The aim is to collect different pieces of information in the form of disks (see Figure 2). These disks contain different information that is needed to escape the planet. The difficulty faced is beating the enemies who try to protect and keep the disks. Both the game maps and the playground are generated based on procedural content generation (PCG) [1] algorithms so that they are different every time, which makes the game more re-playable. To enable an appropriate level of difficulty for the players' skills, dynamic game balancing (DGB) techniques are used. According to the predefined difficulty level, the players have to defeat the enemies and collect all the disks. At the end of each successfully passed mission, a new concept is presented in a textual way. These contents can be customized for each mission and level of difficulty, so educators can determine the value of a single level as also respectively that of a learning concept. To guarantee a high learning success level the players must answer a question according to the previous concept. If it is answered correctly the mission qualifies as passed, otherwise the whole level must be repeated until the question has been finally mastered.



Fig. 2. Player has to collect disks in the explorative gaming mode.

Practical mode: In this mode, the players have to apply the previously learned concepts. The playground is a squared chessboard in which a robot is placed (see Figure 3). The aim is to navigate the robot over the field to reach the disks and avoid obstacles. With the aid of different commands, the player can control the robot and navigate over the playfield. The interface consists of three different tabs: the game instructions, a code editor and an output panel. The goal of the mission is given in the game instruction tab. It also provides the player with all necessary information. The code editor is the place where the actual game takes place. The player can drop different commands in the editor that the robot executes. The commands are displayed as different blocks that can be dragged to the editor. Each block represents an instruction in Python. The game environment provides a Python interpreter (IronPython [18]), where all commands can be executed. The code blocks can be classified in four different sections: print command, variable declaration, move commands (left, right, down, and up), and control structures (conditions and loops). These blocks can be enabled or disabled in the web application, and thus different blocks are displayed depending on the mission. The commands can be changed via a virtual keyboard, so different input or output can be generated. The output panel displays all messages that are dumped from the robot. It also informs the player about the occurrence of syntax errors. Depending on to the player's progress (pre-defined in the web application) the disk is placed on the 15x15 map. The field is divided into thirds each of a different level (50%, 70% and 100% difficulty).

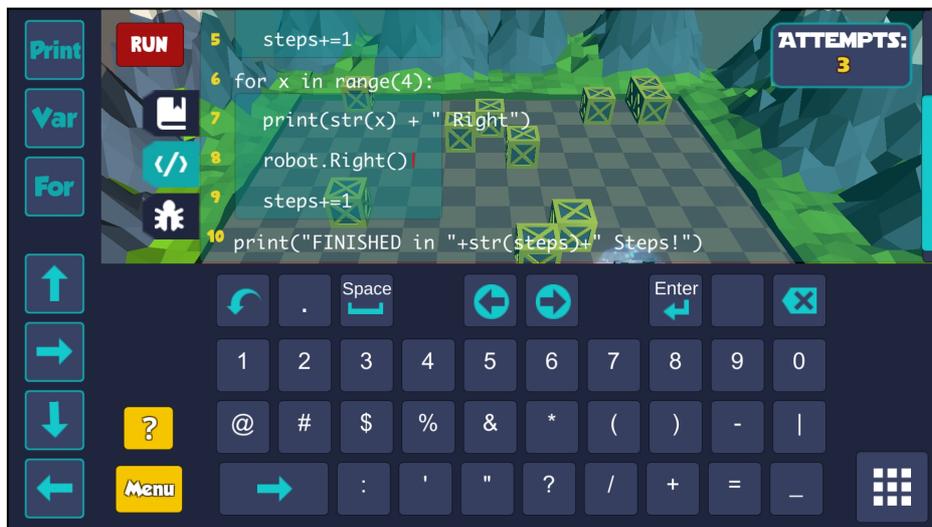


Fig. 3. Players can drag and drop commands to steer the robot.

3.2 Web platform

The .NET-based web application is a specially designed tool for educators. Teachers can use it to provide new educational contents, create new courses or edit existing

courses. Each course is represented as hierarchical skill-tree. These skills are superior elements for all according to concept-learning and practical tasks. Different concepts can be mapped to a specific skill-tree. The content and the degree of difficulty can be declared in the web application and the maps are generated based on these parameters. A distinction is also made between concept-learning and practical tasks on the web platform. In the concept-learning part educators can define learning contents and the corresponding single choice question for the players. Additionally, the difficulty level for the exploration mode is appointed, so the map is generated according to that value. For the practical tasks, educators can define the goal of the task and its reference output. This solution represents the expected output of the Python code to successfully finish the level. In this section it is also possible to define the accessible blocks/commands in the levels.

The web platform also provides an assessment and analytics tool for educators to analyse the learning process of the students and for gaining a detailed evaluation of the course. The mobile app communicates with the server via a JSON API. It fetches the provided course contents and also sends user-related data for analysis purposes.

4 Experiments in Secondary Schools

The main interest of the study was to evaluate if the pupils can learn different concepts in computer science with the aid of the mobile game. We thus tried to establish the engagement and motivation of the class in the context of the mobile game and STEM-learning. We worked with two classes (see Table 1) of different school types (new secondary school and academic secondary school) as part of the computer science class. There were 18 learners in the first experiment (11 girls and 7 boys) of the ages 12-14 years. The second experiment was in a 4th class of an academic secondary school with 12 pupils (6 girls and 6 boys) of the age 13-15 years. Both school classes had no further experience in programming, except for three pupils in the academic secondary school who stated that they had a little elementary knowledge in Java or Scratch. The pupils of the academic secondary school had also worked with different game-based learning approaches in previous school lessons, so they were familiar with different concepts (mobile learning, competitive gaming context, collaborative skills, etc.).

Table 1. Information about the participants in the experiment.

	Experiment 1	Experiment 2
School type	new secondary school	academic secondary school (lower cycle)
Grade	7th	8th
Number	18 pupils (11 girls, 7 boys)	12 pupils (6 girls, 6 boys)
Age	Age 12-14 (M=12.72, SD=0.73)	Age: 13-15 (M=13.75, SD=0.59)
Programming experience	none	3 pupils

4.1 Procedure and instruments

In each class the experiment lasted 100 minutes (two lessons á 50 minutes). At the beginning we briefly introduced the mobile game by telling a story about a robot named Rob who was on a space discovery mission and suddenly had to make an emergency landing on a distant planet.

The next step was to build groups of two (each group should have an Android device) and install the app via a provided link. Once the game was installed on each device, we handed out a paper with the first instructions: The students had to accomplish five given planet exploration missions (concept-learning part). After they solved these levels successfully, they had to solve three given robot missions. Each level was based on the previous one and the whole course had a coherent narrative, where the players had to follow the story line and help a robot to collect all the pieces of information needed to escape from the distant planet.

The groups, which finished within 50 minutes, were handed out a worksheet (see Figure 4) with a simple problem that builds on the course contents. The task was to print the three times table (3 to 30) within a single print command using loops.

After 50 minutes, the class had to fill out the evaluation forms. At the end of the lesson, we asked the students about improvements and their opinion about the game. To do this we provided a questionnaire on Google Forms [19] with different questions. First, we asked about demographic information and mobile learning experience. We draw on the game engagement questionnaire [20] to evaluate engagement of the players while playing the game.

Another interesting aspect was that of the emotions the students experienced while playing the game. We used the 12-item computer emotion scale [21] questionnaire for this purpose. For assessing the gender-specific perception of the girls we asked them five rating questions (Likert scale). At the end of the questionnaire, we provided an open-ended field for any additional comments and input.

All responses to questions based on statistical evaluable data (Likert scale, demographic data, etc.) were analysed by using R 3.5.1 [22]. The response of the open-ended questions was categorised in Microsoft Excel 2016.

school name: _____

Rob the Roboter and the delayed departure



Ready for take-off in ...

Now it's done! You finally helped Rob to find all missing disks, so he can fly back to the earth – congratulation! Unfortunately, there was an error while reading the disks and so the start sequence couldn't be loaded correctly – it seems like Rob cannot start. Apparently, there is a problem with the maths module that is needed for the flight back. Rob has to do different calculations but isn't possible to solve them. He needs a program that performs different multiplications. The broken program was responsible for calculating the three times table. Can you help Rob to re-write the program, so he can finally escape from the planet and take back home all collected research results?

Output

The output should look like this:
3, 6, 9, 12, 15, 18, 21, 24, 27, 30 – Calculating finished!

Code

Hint

You can use the commands that you have already learned during your space missions (variables and loops).

Fig. 4. Worksheet with the final programming task.

4.2 The missions

We created a new course for the mobile game named “Basic Coding” in accordance with the age of the students. Before we carried out the experiments, we first inquired with the computer science teachers, if the pupils have any previous knowledge in coding to which the course should be adequately adapted. Since none of the students had any further experience in programming, we decided to make a start with the fundamental concepts.

The five concept-learning tasks (see Table 2) were divided into a “Hello World” example, where the players learned how the robot can print a specific output. The following three levels were about variables, so the students first learned how to store a text in a variable, how basic arithmetical operations work plus the possibility for stor-

ing the result in a variable and subsequently they learned how to print the content of a variable. The last introduced concept was that of the loops, in which they learned how to repeat a specific instruction a given number of times.

All three of the concepts learned have been applied in a practical manner based on the explorative missions in the narrative. Apart from leading the robot to the disk, there were some additional tasks. The first practical task was based on the first concept-learning mission where the print command was introduced. The goal was to reach the disk and print the name of the robot. In the second mission, the players had to do a very basic calculation, store the result in a variable and print it before they could collect the disk. We limited the given coding components for these two tasks to the steering (back, forward, left and right), printing and the variable blocks. The last task was about using loops: The students had to count from 20 to 30 and reach the disk. In this mission, they were also provided with the loop block. The difficulty of the level was set to 60% in the web application, so the disk was placed in the second third of the playfield. As a result, the students had to use the steering blocks very frequently. The idea was to guide the player in this way to use loops instead of rewriting the same command each time.

Table 2. Summarize the missions in the experiments.

Mission	Concept	Difficulty
Concept-learning task 1	Print	20%
Concept-learning task 2	Variables, Strings	30%
Concept-learning task 3	Variables, Arithmetic	40%
Concept-learning task 4	Print, Strings, Variables	50%
Concept-learning task 5	Print, Loops	60%
Practical task 1	Print	20%
Practical task 2	Print, Variables, Arithmetic	40%
Practical task 3	Print, Loops	55%

4.3 Evaluation methods

After the practical part the students were asked to fill in answers for a questionnaire asking about their demographic information and their experience with mobile games and game-based learning. The questionnaire included the game engagement questionnaire (GEQ) [20] and the computer emotion scale (CES) [21], and some open-ended questions about the game and the game experience. A particular field of interest was the play experience of the girls. The female participants were thus asked additional questions about the game mode. They also had a free text question for individual responses.

Computer emotion scale: The computer emotion scale [21] is a survey with 12 different items that are associated with four basic emotions (happiness, sadness, anxiety, and anger). On a Likert scale from 1 to 4 the participants have 9 specific points to report on how they felt while playing the mobile game.

Game engagement questionnaire: The game engagement questionnaire (GEQ) [20] consists of 19 items that can be assigned to the players concerning the engagement they experience when playing the game. The questionnaire distinguishes be-

tween four different dimensions: immersion, presence, flow, and absorption. A Likert scale between 1 and 5 is used to describe the states.

Worksheet: A special worksheet was prepared for assessing whether the students can assign the concepts they have learned to other given problems in a similar domain. The prerequisite was to complete all concept-learning and practical missions in the mobile game in order that all the necessary concepts are known. The worksheet was embedded in the whole game story where the students have to help the robot to escape from the planet. In this context they had to fix the broken “calculation module” by writing a piece of code that counts from 3 to 30 in steps of 3. The objective of this exercise was to find out if the students can transfer the concepts of loops and variables from the domain of the space mission to a that of mathematical field. They were provided with an additional hint about the concepts to be used.

4.4 Experiment results

Since there are some differences regarding experience in game-based learning we decided to separate the evaluation of the game engagement questionnaire and the computer emotion scale according to the school types.

After playing the game, the students of both groups were asked to fill in the game engagement questionnaire, to evaluate their engagement. In the new secondary school (see Figure 5) engagement was significantly better than in the academic secondary school (see Figure 6). According to the immersion ($M=3.78$; $SD=1.47$) the players of the first group were profoundly caught up in the game. By contrast, the second group was less immersed ($M=1.83$; $SD=1.34$). An important factor concerning motivation and learning is the flow [8]. The first group had a relatively high level of flow ($M=3.53$; $SD=1.5$) compared to the second group ($M=1.79$; $SD=1.26$).

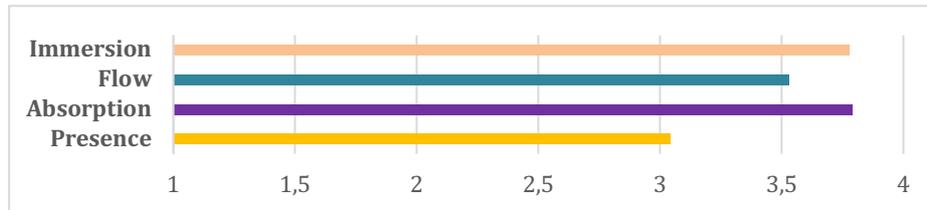


Fig. 5. Game engagement in new secondary school pupils.

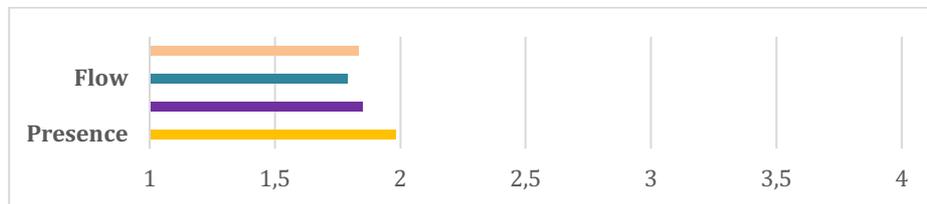


Fig. 6. Game engagement in academic secondary school pupils.

The second factor to be evaluated was that of emotions [21] experienced concerning the game. In both groups the game caused mainly positive emotions. Bad emotions such as anxiety, sadness or even anger had a relatively low occurrence in both groups. But the happiness level ($M=3.39$; $SD=0.63$) in the first group (see Figure 7) was higher than in the second group (see Figure 8) ($M=2.56$; $SD=0.97$).

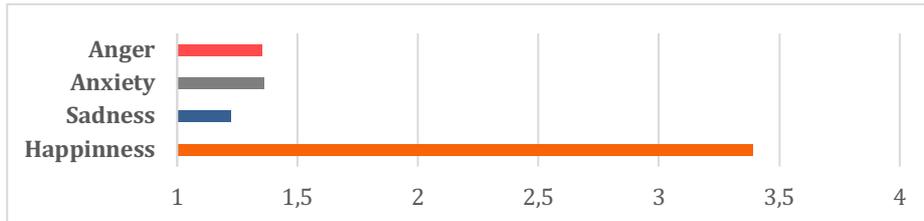


Fig. 7. Emotions when playing the game in new secondary school pupils.

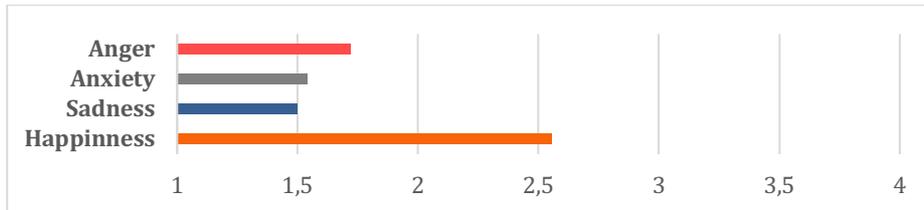


Fig. 8. Emotions when playing the game in academic secondary school pupils.

All 17 groups were able to pass every concept-learning game mission within the given time. The average duration for solving a single level was 0.96 minutes. Only 35.75% passed the levels on their first try. On the one hand this was caused by the fact that the game enemies defeated the players but on the other hand the players did not read the text carefully and were thus unable to give the right answer. After a few tries the pupils realized that they would not be able to pass the level when they skipped the learning contents. On comparing both school types, the academic secondary school (0.75 minutes) fares a little better than the new secondary school (1.18 minutes).

After accomplishing all concept-learning tasks, the players started with the practical missions. 15 out of 17 groups were able to solve all three practical missions. The average duration was 4.82 minutes per level. In the third mission there were two alternative possibilities to the use of the loop block. First, the level of difficulty was chosen above 50%, so the disk was placed in the second third to engage players in using loops instead of using the same control block each time. The second scope was to count from 20 to 30. Three out of 15 groups in this level did not use loops for counting to 30. They used a single print command with the expected output instead. No group used the loop block to repeat a moving command. By comparing the average time per mission, the academic secondary school pupils were about 27% faster (4.07 minutes) than the new secondary school pupils (5.57 minutes).

At the end of the experiment 9 groups tried to solve the given worksheet. The pupils were asked to write the Python code on paper. We told them there was no need for a correct syntax emphasis and that they could also use the game. After a few minutes we collected all worksheets and wanted to discuss the solutions together. One single group only had a broad idea of how to solve the task with a loop. A second group used a print command, but none of the other groups could not find an appropriate solution.

At the end of the lesson the students filled in a questionnaire. One item was an open-ended question about the game in general. The feedback can be classified into two groups: language and game control. Five (out of 30) students said that the language used (English) was too difficult to understand and they asked for a German game version. The second category was about the control. The pupils answered that they had experienced difficulties in steering the robot.

4.5 Gender-specific aspects

A thematic priority of our experiments was the perception of the female participants. We thus offered five additional questions (see Table 3) and some open-ended questions to cover this issue. These five questions were rated on a five-element Likert scale.

Table 3. Answers of the questions about the girls’ perception.

	Strongly agree	Agree	Undecided	Disagree	Strongly disagree
I liked the kind of game.	11	5	1	0	0
The character ‘Rob the Robot’ was appealing.	10	5	1	0	1
I enjoyed playing the game.	11	4	1	1	0
The topic ‘space’ is interesting.	6	3	5	2	1
I enjoyed programming.	7	5	3	1	1

As shown in Table 3, most of the girls enjoyed playing the game and its space theme. In the open-ended questions, only one girl from the entire group, pointed out that there is no possibility to change either the sex or the name of the robot. Apart from this, all other answers were related to the game itself (control, missions and user interface).

After 50 minutes six out of nine of the girl groups finished all the concept-learning and practical levels. At the same time, four out of seven of the boy groups solved all the tasks. Compared to the average duration of the explorative missions the girls (0.87 minutes) were very similar to the boys (0.83 minutes). In the practical missions, the girls (4.31) were even slightly faster than the boys (4.40 minutes). The girl groups were able to complete the concept-learning levels with a 27% success rate at the first try. The boys solved these missions at the first time with a 44.29% success rate. The average number of attempts at the practical missions (number of restarts at the robot mission) was very similar for both groups. All gender-related results are listed in the subsequent Table 4.

Table 4. Comparing the performance categorized by players sex.

Sex	Groups	Groups completed	Concept-learning		Practical	
			Avg. Duration	1st attempt	Avg. Duration	No. of attempts
Female	9	6 (66.67%)	0.87 min	27%	4.31 min	2.50
Male	7	4 (57.14%)	0.83 min	44.29%	4.40 min	2.43
Mixed	1	0	2.83 min	57%	-	-

5 Findings and Conclusion

We extended the study of the educational videogame sCool and its usage in school classes in this study based on our previous work [1, 2]. In two experiments with the same course set-up but different school types, we wanted to determine if sCool can be used in an educational context to improve the students’ computational thinking skills.

The evaluation showed that the students in the first group (new secondary school) were more engaged. This can be traced back to the fact, that the second group had much more experience with game-based learning in class, while it was a first experience for the other groups. In a comparison of the performance of both school types, the academic secondary school fared slightly better. They passed through both the concept-learning and the practical missions faster. After the course in the video game the groups should be able to solve a practical worksheet, based on the learned concepts. No group could transfer the concepts from the game to a similar task.

In addition to the evaluation of the performance of all the players we also analysed the perception of the game by the girls. Most of the girls liked the game and its topic. The majority also enjoyed coding the robot. One girl mentioned some gender-specific issues with the game respectively the story. The performance of the girls was similar to that of the boys and they were faster than the boys in the practical missions.

For future work, we intend to adapt the design of the courses, so the abstraction from the concept in the game to a real-life example will be more of a natural step. With this intention we are reworking the practical missions and adding more possibilities for students to gain a better understanding of certain specific concepts. We also will focus on the gender aspect and rethink both the character of the game and the outer skin it presents, to make it more attractive to girls.

6 Acknowledgement

We wish to thank Aleksandar and Milos Kojic from Graz University of Technology as well as Daphne Economou and Markos Mentzelopoulos from Westminster University for their great effort in designing and developing the basic version of sCool [1]. We also wish to thank Roland Miniberger and his pupils from the New Secondary School Steinerkirchen and Gerhard Fetka and his pupils from the Academic Secondary School Kirchengasse for taking part at the experiments.

7 References

- [1] Kojic, M. Kojic, J. Pirker, C. Gütl, M. Mentzelopoulos and D. Economou, “sCool – A Mobile Flexible Learning Environment,” in Online Proc. from 4th Immersive Learning Research Network Conference, Missoula, Montana, 2018.
- [2] A. Steinmaurer, J. Pirker and C. Gütl, “sCool – Game Based Learning in STEM Education: A Case Study in Secondary Education,” *Advances in Intelligent Systems and Computing*, 2018.
- [3] J. M. Wing, “Computational thinking,” *Communications of the ACM*, vol. 49, no. 3, pp. 33-35, 03 2006.
- [4] A. Dasgupta, A. M. Rynearson, W. Lafayette, S. Purzer, H. Ehsan and M. Cardella, “Computational Thinking in K-2 Classrooms: Evidence from Student Artifacts (Fundamental),” in 2017 ASEE Annual Conference & Exposition, Columbus, Ohio, 2017. <https://doi.org/10.18260/1-2--28062>
- [5] J. L. Plass, B. D. Homer and C. K. Kinzer, “Foundations of Game-Based Learning,” *Educational Psychologist*, vol. 50, no. 4, pp. 258-283, 2015. <https://doi.org/10.1080/00461520.2015.1122533>
- [6] A. Bartel and G. Hagel, “Engaging Students with a Mobile Game-Based Learning System in University Education,” *International Journal of Interactive Mobile Technologies*, vol. 8, no. 4, 2014.
- [7] O. Mironova, I. Amitan and J. Vilipöld, “Programming Basics for Beginners. Experience of the Institute of Informatics at Tallinn University of Technology,” *International Journal of Engineering Pedagogy (iJEP)*, vol. 7, no. 4, pp. 7-18, 2017. <https://doi.org/10.3991/ijep.v7i4.7425>
- [8] R. Hunicke and V. Chapman, “AI for dynamic difficulty adjustment in games,” *Challenges in game artificial intelligence AAAI workshop*. 2, 2004.
- [9] J. Pirker, C. Gütl and Y. Astatke, “Enhancing online and mobile experimentations using gamification strategies,” 2015 3rd Experiment International Conference (exp.at'15), pp. 224-229, 2015.
- [10] A. Yassine, M. Berrada, A. Tahiri and D. Chenouni, “A Cross-Platform Mobile Application for Learning Programming Basics,” *International Journal of Interactive Mobile Technologies*, vol. 12, no. 7, 2018.
- [11] Z. Karabatzaki, A. Stathopoulou, G. Kokkalia, E. Dimitriou, P. I. Loukeri, A. Economou and A. Drigas, “Mobile Application Tools for Students in Secondary Education. An Evaluation Study,” *International Journal of Interactive Mobile Technologies*, vol. 12, no. 2, 2018. <https://doi.org/10.3991/ijim.v12i2.8158>
- [12] M. Romero, M. Usart and M. Ott, “Can Serious Games Contribute to Developing and Sustaining 21st Century Skills?,” *Games and Culture*, vol. 10, no. 2, pp. 148-177, 2015. <https://doi.org/10.1177/1555412014548919>
- [13] A. Gomes and A. J. Mendes, “Learning to program - difficulties and solutions,” in *International Conference on Engineering Education – ICEE 2007*, Coimbra, Portugal, 2007.
- [14] T. A. Lin, “Playful Coding: Game-Based Teaching for Programming Courses,” in *The Asian Conference on Technology in the Classroom 2016: Official Conference Proceedings*, Kobe, Japan, 2016.
- [15] “CodeMonkey,” CodeMonkey Studios Inc, [Online]. Available: <https://www.playcodemonkey.com/>. [Accessed 16 12 2018].
- [16] J. Ashkenas, “CoffeeScript,” [Online]. Available: <https://coffeescript.org/>. [Accessed 16 12 2018].
- [17] “Lightbot,” LightBot Inc, 2017. [Online]. Available: <http://lightbot.com/>. [Accessed 16 12 2018].

- [18] “IronPython,” .NET Foundation, [Online]. Available: <http://ironpython.net/>. [Accessed 16 12 2018].
- [19] “Google Forms,” Google LLC, [Online]. Available: <https://www.google.com/intl/en-GB/forms/about/>. [Accessed 16 12 2018].
- [20] J. H. Brockmyer, C. M. Fox, K. A. Curtiss, E. McBroom, K. M. Burkhart and J. N. Pidruzny, “The development of the Game Engagement Questionnaire: A measure of engagement in video game-playing,” *Journal of Experimental Social Psychology*, vol. 45, no. 4, pp. 624-634, 2009. <https://doi.org/10.1016/j.jesp.2009.02.016>
- [21] R. H. Kay and S. Loverock, “Assessing emotions related to learning new software: The computer emotion scale,” *Computers in Human Behavior*, vol. 24, no. 4, pp. 1605-1623, 2008. <https://doi.org/10.1016/j.chb.2007.06.002>
- [22] “The R Project for Statistical Computing,” The R Foundation, [Online]. Available: <https://www.r-project.org/>. [Accessed 16 12 2018].

8 Authors

Alexander Steinmaurer is studying in the Teacher Training Programme on the subject Computer Science on Graz University of Technology. He is currently working on his master's thesis about game-based learning in STEM education. His research focus is on game-based-learning, gamification and knowledge transfer (email: a.steinmaurer@student.tugraz.at).

Johanna Pirker is a computer scientist focusing on game development, research, and education; she is an active and strong voice in the local indie dev community. She has lengthy experience in designing, developing, and evaluating games and VR experiences and believes in them as tools to support learning, collaboration, and solving real problems. Johanna has started in the industry as QA tester at EA and still consults studios in the field of games user research. In 2011/12 she started researching and developing VR experiences at Massachusetts Institute of Technology. At the moment she teaches game development at TU Graz and researches games with a focus on AI, HCI, data analysis, and VR technologies. Johanna was listed on the Forbes 30 Under 30 list of science professionals. (email: jpirker@iicm.edu).

Christian Gütl is Associate Professor and based at the Institute of Interactive Systems and Data Science at TUG in Graz, Austria, where he leads the Motivational Media Technologies (MMT) Group. Christian has authored and co-authored more than 200 peer-reviewed book chapters, journals, and conference proceedings publications. He is founding member of the global Immersive Learning Research Network (iLRN), managing editor of J.UCS, co-editor of the International Journal of Knowledge and Learning (IJKL), and general chair of IEEE SNAMS. His research interests include information search and retrieval, natural language processing, social media technology as well as e-education, e-assessment, and adaptive media technologies and virtual and augmented reality for learning and knowledge transfer. (email: c.guetl@tugraz.at).

This article is a revised version of a paper presented at the International Conference on Interactive Collaborative Learning (ICL2018), held September 2018, in Kos, Greece. Article submitted 2018-11-30. Resubmitted 2019-01-29. Final acceptance 2019-01-29. Final version published as submitted by the authors.