DOI: https://doi.org/10.3217/978-3-85125-609-3-11

sCool - A Mobile Flexible Learning Environment

Aleksandar Kojic(✉)[1], Milos Kojic[1], Johanna Pirker[1], Christian Gütl[1,2], Markos Mentzelopoulos[3], and Daphne Economou[3]

[1] Graz University of Technology, Austria
{m.kojic@,a.kojic}@live.com,jpirker@iicm.edu
[2] Curtin University, Perth, Western Australia
cguetl@iicm.edu
[3] University of Westminster, UK
{M.Mentzelopoulos01,D.Economou1}@westminster.ac.uk

Abstract. We present sCool, an adaptive mobile-based educational game designed for STEM education. sCool is designed as a two-fold system. The first part is designed for learners. It is a mobile game supporting modes for explorative knowledge collection and also hands-on practical learning. This part is designed to be highly flexible and adaptive to support different play styles, learning habits, as well as course programmes. The second part is designed for instructors and supports a web-based interface to create new courses and assess the learners' performance. In a preliminary study, we found that learners were engaged by this form of learning, but that different form of gameplay should be supported to engage different target groups. In a first study with teachers, the usability was rated as very high and elements such as adaptability of the course material, as well as assessment and analytics of learners' behavior, are important elements of such platforms.

Keywords: Gamification · Educational games · Game-based learning

1 Introduction

Our modern society moves very fast, and new ways of collaboration and working as well as technological inventions requires new and adapted skills and knowledge. STEM (science, technology, engineering and math) fields become increasingly important in this context. Former US present Obama emphasized the importance of STEM as it is " *[...] more than a school subject, or the periodic table, or the properties of waves. It is an approach to the world, a critical way to understand and explore and engage with the world, and then have the capacity to change that world and to share this accumulated knowledge. Its a mindset that says we that can use reason and logic and honest inquiry to reach new conclusions and solve big problems* [1]." There is, however, a lack of interest in STEM topics by students in schools and at universities [2, 3]. This situation calls for further initiatives and support from policy makers, researchers, and educators.

Our modern society has also changed how our children communicate, learn, and play. This new generation, also termed Generation NeXT, grown up or grew

DOI: https://doi.org/10.3217/978-3-85125-609-3-11

up with new technology, mobile devices, social media and video games. This environment has changed their ways to communicate, collaborate, and learn. They are seen as prosumer of information but also consume information in smaller portions. They are used to use a variety of sources and media and tend to be multi-taskers using their preferred apps on their mobile devices. Consequent, the Generation NeXT calls for adapted ways how to learn and acquire skills, such as interactive, engaging, and playful micro learning activities accessible by mobile devices [4].

The situation stated so far has motivated us to initiate a research project focusing on a mobile game for razing attention of and teaching STEM topics, in particular computational skills. Educational games and applications, however, tend to support only static content and instructors often have issues adding or changing learning content. Additionally, these games and applications often do not provide any or sufficient assessment options to gain an understanding of the students learning performance, progress, and status. An adaptive and flexible system, which is designed for instructors but connected to the application for students and supporting options such as adding, changing, and removing content, as well as learner assessment would address this issue. In the light of this, the aim of our initiative is to research, prototype, and evaluate a flexible mobile learning tool for STEM education. Thus the main contributions of this paper can be summarized as follows:

- Design and implementation of sCool, an adaptive mobile learning tool supporting different courses and play styles
- Demonstration through a preliminary study that learners experience engagement while learning with this game
- Demonstration through a preliminary study that teachers would use tools to create own courses or course items for an educational mobile game.

2 Related Work

Playing is a natural form of learning and training skills [5], which also has been transferred into the video games domain since many years [6]. Unlike other learning and training scenarios, game-based approaches incorporate aspects such as trial and error, repetition, and improvements as well as fun and reward. Mobile devices are becoming the most important tool for the younger generation [7]. This generation NeXt introduced also the trend of micro learning [8], where smaller units of knowledge are mastered in a mix of different learning activities.

There is a general shortage of STEM (science, technology, engineering, and math) graduates and a lack of interest in related topics worldwide, and in addition, there is still a big gender gap in STEM topics disadvantaging women. To overcome these issues, there are many STEM education initiatives on a national and an international level. (Murray, 2017; UNESCO, 2017). STEM education has also raised increasing attention over the last years in the e-learning domain to support a great variety of subjects, such as game-based and mobile game-based learning [9].

DOI: https://doi.org/10.3217/978-3-85125-609-3-11

Based on the situation outlined above, interest for and motivation to learn STEM topics for the generation NeXT can be raised by mobile game-based approaches, which offer small learning experiences in a highly interactive way. In one illustrative example, [10] offered a mobile learning experience to playfully acquire historical knowledge. They could show higher engagement and motivation to learn and significantly higher knowledge acquisition over 458 pupils from 20 classes. An overview of mobile learning trends in this context is given in [11].

Narrowing down to the application domain covered in the research of this paper, there are various research and development initiatives to support computation thinking and to teach coding. MITs Scratch [12] enables children create their own interactive stories, games, and animations using a visual programming language on the desktop. In a very similar way transfers Catrobat [13] the approach to mobile devices. Programmable robots can also engage children and students to learn coding and master challenges [14].

Many game-based approaches also focus on teaching computational thinking and programming. There are desktop programs, such as Robocode[4] which teaches JAVA and .NET by programming a vehicle to fight against enemies, or CodeSpell[5] which uses drag-and-drop block language the create magic spells and apply it in the game environment. Code Combat[6], a Web-based game, requires payers to type in a code in order to control the character and fight the enemies. Code Hunt [15] and CodinGame[7] are also Web-based games teaching JAVA, .NET, or scripting languages, which are optimized for desktop usage. However, most of these tools are designed as web- or desktop-applications. In order to overcome this issue, some programming games are specifically designed for mobile devices. Swift Playgrounds [8] is an iPad game for teaching the programming language Swift to control and move the character in the game. Tynker and Box Island are also games for tablets and teaching introductory computer science concepts [16]. SpriteBox[9] is a mobile game with a mixture of exploration and learning kids to code. It teaches basic principles of programming by exploring knowledge, solving logical puzzles and filling in missing parts of the code.

Video games have shown a great pedagogical potential and this is one of the reasons why there are more and more game-based learning tools. Large-screen devices are already being used by broad masses in educational purposes and these devices have enough space for displaying different types of educational content. So the majority of educational tools for teaching both basic and advanced concepts are being dedicated to desktop and tablet devices. Even though mobile devices are increasingly trending, there is an obvious lack of mobile educational tools for teaching more advanced concepts for high school and first-year university students [17]. Some existing mobile games for learning programming

[4] robocode.sourceforge.net
[5] codespells.org
[6] CodeCombat.com
[7] CodinGame.com
[8] https://www.apple.com/swift/playgrounds/
[9] SpriteBox.com

DOI: https://doi.org/10.3217/978-3-85125-609-3-11

are only based on teaching concepts of programming logic. There might be an increasing demand for more educational tools of this kind as the number of mobile users is rapidly increasing. Following, we introduce sCool. sCool simulates a real programming environment, adapts the educational content and the game difficulty to the players needs and offers a web platform for content creation.

3 System Overview: sCool

Even though mobile devices are increasingly trending, there is a lack of mobile game-based educational software for teaching more advanced concepts for high school and first-year university students. A problem which educational games usually encounter is the complexity of level creation. Instead of focusing on educational content creation, educators have to invest a lot of effort in designing every single level. A proposed solution for this problem is based on utilizing procedural content generation (PCG) algorithms for generating terrains, maps, content, and sounds. Adaptive learning and dynamic game balancing (DGB) principles ensure that both the educational content and the game are adapted to a user's ability levels. Therefore, the sCool project was created as a solution for increasing motivation and engagement of students in the age 13-19 in the fields science, technology, engineering, and mathematics (STEM). A programming course is fully implemented and was used as a case study while other courses are in the development process.

The implementation section consists of the implementation of the two main components which are the mobile video game and the web platform. The 3D mobile game was designed as a learning tool for students. It displays the educational content in a form of challenges and quizzes. It was developed with the Unity3D game engine. The web platform provides an interface for educators to easily create courses, add educational content, and track the students' progress. Educators are able to create courses with corresponding categories and subcategories for both practical and theoretical aspects. The final course represents a knowledge tree which is loaded into the game. The knowledge tree for the Python programming course consists of skills such as declaring variables, using functions, branches, and loops. Once students select a course, they are provided with lessons, instructions, and finally tasks which they have to solve. The mobile game utilizes the servers web APIs to obtain the content and afterward sends the results and the statistics back to the server. Also, the web platform provides instructors with tools for leaner analytics.

3.1 Web Platform

When creating a course, educators have to provide educational content for both theoretical and practical modes. In the theoretical mode, educators provide a lesson in the form of text and a quiz with correct and incorrect answers, while the practical mode consists of tasks which are defined by the task description and expected output values. In the first prepared course, a programming course

DOI: https://doi.org/10.3217/978-3-85125-609-3-11

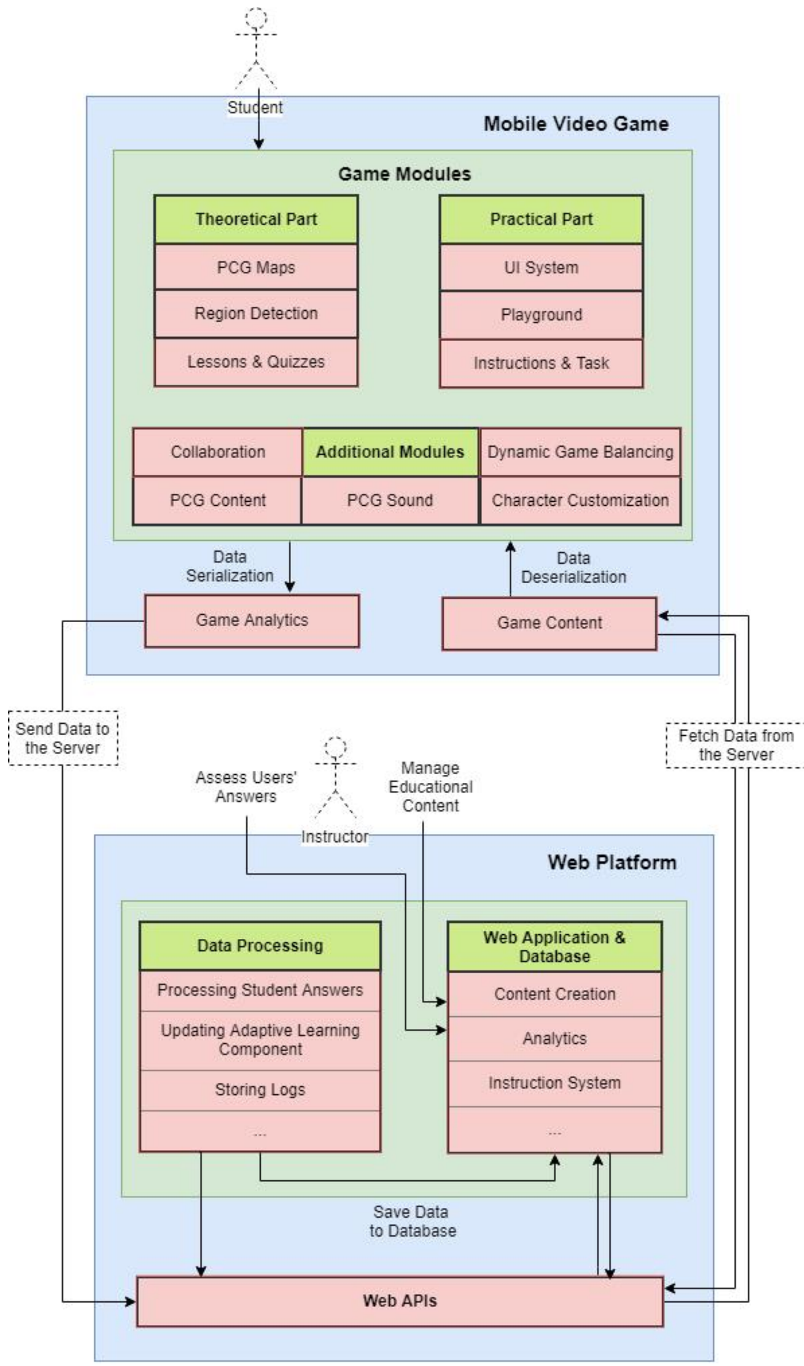

Fig. 1: The sCool game - Architecture.

DOI: https://doi.org/10.3217/978-3-85125-609-3-11

featuring Python, students are able to learn more about Python and afterward to type code, solve given task, print results to the output, and compare them with the correct results provided by educators.

The web platform is based on the client-server architecture. The architecture of the sCool project, from the technological perspective, is presented in Figure 1. Every course consists of both theoretical and practical modes, therefore, the process of learning in the game is very similar to a classroom experience. One of the main benefits of the way the content is structured in the game is that the students can see the structure of the entire course, their current progress within the course, as well as, the progress of their classmates.

The web application was built in the ASP.NET MVC web framework. Entity Framework (EF), an object-relational mapper, is utilized for facilitating the data access. By using the EF, the process of code writing was easier as well as the data manipulation. Based on the entity classes, front-end files with Razor syntax are created for generating HTML output. The implemented APIs are based on the ASP.NET Web API framework. This framework offered the Representational State Transfer (REST) interface which served as an access point for a data sent to and received from the mobile game. For serialization and deserialization purposes, Json.NET is utilized, which is a high-performance JSON framework for .NET. After the implementation process was completed, the web application was hosted on the Microsoft Azure cloud platform. A screenshot of the web platform is shown in Figure 2. There is a number of analytics tools which educators can use for tracking individual and group performances as well as to identify weak points and provide support for improving them. The analytics tools are based on MVC components and UI libraries such as Bootstrap[10], Morris.js[11], Flot[12], and others.

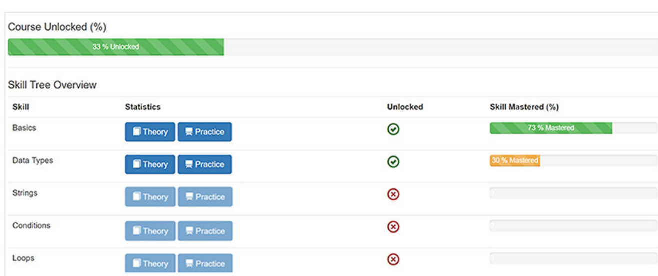

Fig. 2: The sCool web platform.

[10] https://getbootstrap.com/
[11] http://morrisjs.github.io/morris.js/
[12] flotchars.org

DOI: https://doi.org/10.3217/978-3-85125-609-3-11

3.2 Mobile Game

This section consists of the implementation of the theoretical and practical modes of the mobile video game. Both of these components require educational content which needs to be retrieved from the server and deserialized before a level starts. The purpose of the game is to help educators focusing on the educational content creation rather than a level, map, or world creation. In order to achieve that it was necessary to use PCG techniques [18].

Theoretical mode - In the theoretical mode, players have to explore maps, defeat enemies by shooting at them, pick up required pieces of information, and provide correct answers to questions which show up at the end of each level. Map structures and terrains are created based on PCG algorithms (Cellular Automata and Perlin noise) which ensure that on every run, players get a different environment [19]. Cellular Automata algorithm was used for generating cave-like shapes and making a distinction between walkable and non-walkable areas. Since generated maps were represented with matrices, it was possible to use those matrices for mesh generation and applying noise to certain parts of the maps. Perlin noise was utilized to ensure that non-walkable parts of maps are transformed into natural-looking terrains. The tree and sound generation was implemented in both modes. The final part of the procedural map generation process was to place game objects such as the player, enemies, obstacles, and pickups on the map. It was necessary to design an algorithm which goes through walkable area and searches for spots where those objects can be placed [20]. A final outcome of the map generation process is shown in Figure 3.

Fig. 3: The sCool game - PCG maps.

Practical mode - In the practical mode of the programming course, the goal is to follow instructions specified by educators, program a robot to avoid obstacles, and collect a disk placed on a map. The obstacles are procedurally created and spread out the map. Two most important components of the practical part are a UI system and an environment (playground). The UI system consists of tabs,

DOI: https://doi.org/10.3217/978-3-85125-609-3-11

code blocks, fields, virtual keyboard. Code blocks contain code snippets which generate code when dragged and dropped onto the programming area. Custom Fields were built to replace input fields, which are not convenient for typing the code as they cannot be sorted or nested. The custom virtual keyboard improves the process of typing the code and provides the majority of functionalities which one can find in default mobile keyboard. The most significant component in the playground is Python code execution. That was achieved through IronPython which provides error reporting and allows execution of Python code on the .NET platform. This means that the code which users provide has to be valid otherwise they get an error (see Figure 4)

Fig. 4: The sCool game - programming tools.

Dynamic Game Balancing (DGB) - In both theoretical and practical modes, DGB principles are utilized in order to keep players in the flow channel where the game is neither too challenging nor too easy. Therefore, a DGB approach of Hunckie and Chapman[21] is used. This DGB approach is utilized in a way of adjusting game parameters such as health points, damage value, attack speed, number of enemy units, map size, and other parameters based on the player's progress. If a player is not performing well, then a map becomes smaller, the number of enemies is decreased, and other parameters become respectively adjusted. The procedure is similar to the opposite case as well to avoid boredom and keep players engaged.

4 Evaluation

The system sCool was evaluated in a two-fold study. The first part of the evaluation focuses on testing the mobile game application with students to measure engagement and observe the interaction with the application. In the second part of the evaluation, the web-based tool to create new courses is tested with the target group (teachers).

DOI: https://doi.org/10.3217/978-3-85125-609-3-11

4.1 Part 1: - sCool for Learners

The educational mobile game was designed for students. The game was designed to generate a high level of motivation and engagement. Thus, this first study was designed to gain insights into the learners' overall engagement and the usability. The learning efficiency was not evaluated in this study.

Methodology and Participants. 12 (2 female) students between 16 and 20 (AVG=18.5; SD=1.24) were recruited to participate in this study. All of them were students from a high school or in their early years at the university. First, they were asked to fill out a pre-questionnaire with the focus on gaining information about their prior experience with educational mobile games and mobile games in general. After that, they were asked to interact with the game. In a post-questionnaire, the learners' were asked to answer different questions with a focus on getting a deeper understanding of their motivation and engagement. The questionnaire included open-ended questions as well as the standardized questionnaire Game Engagement Questionnaire (GEQ) [22]. The Game Engagement Questionnaire (GEQ) is designed as a scale to measure emotions and psychological states important for engagement such as flow, absorption, presence, and dissociation. 16 questions on a Likert scale between 1 to 7 (7 highest) are used to assess these states.

Results. Most participants mentioned that the liked the gameplay. 58% said that they strongly agree that the gameplay was easy to understand. *I am amazed by animations and the overall design. Its easily understandable and appropriate for the audience. By completing a level youre falling deeper and deeper into the world of programming.* Most users mentioned that sCool is a good supplement to regular learning (M=6.0; SD=1.3) and also mentioned that they've learned something new when using sCool (M=5.5; SD=1.38). sCool was also noted as a tool to make learning more engaging (M=5.58; SD=1.24). We used to GEQ to measure engagement based on flow, presence, immersion, and absorption. Absorption was rated with a mean of 1.87 (SD=0.84), flow with 2.38 (SD=1.03), presence with 3.25 (SD=1.22), and immersion very high with a mean of 3.25 (SD=1.22). As this game is designed as a casual-game supporting shooting activities with a gun, we were interested in the perception of the violence in his game. 9 out of 12 mentioned that it is not too violent, 2 agreed, and one was undecided. This part of the game design is definitely a strong factor for future improvements and further discussions.

4.2 Part 2: - sCool for Teachers

The web-application was designed for the target group "instructors and teachers" to support them in creating new courses and adding and editing course content without prior programming skills. Thus, the tool was evaluated with a focus on usability.

DOI: https://doi.org/10.3217/978-3-85125-609-3-11

Methodology and Participants. The web platform is mainly designed for high school and university instructors. They are able to create new courses and course content with this tool. For this evaluation, 10 (2 female) instructors from different universities were asked to evaluate the tool. All of them were already familiar with e-learning tools and the majority of them (60%) had no experience, the rest a little experience with an educational mobile game. In the first step, participants were asked to interact with the web application. After that, they were asked to fill out a questionnaire. The questionnaire was designed with a focus on identifying usability issues and learning how participants learn the new software. The questionnaire included open-ended questions, as well as the two standardized scales (1) System Usability Scale [23] and (2) Computer Emotion Scale [24]. The total duration of the evaluation was 20-30 minutes. The Computer Emotion Scale (CES) [24] is a standardized questionnaire to gain information about the emotion of the users while interacting or learning how to interact with the new software. Thus, the feelings anger, anxiety, happiness, and sadness are measured. These feeling are rated on a scale between 0 and 3. The System Usability Scale [23] is a standardized questionnaire to gain information about the usability of the system. The 10-item questionnaire is based on a 5 point Likert scale and results in a final score in the range of 0-100.

Results. The main emotion described by users was "Happiness". On a scale between 0 and 3 (3 as the highest) the average rating for happiness was 2.08 (SD=0.8), for sadness 0.05 (SD=0.16), anxiety 0.1 (SD=0.24) and anger 0.1 (SD=0.22). The usability was rating very satisfying with a total average score of 84.24 (SD=8.34). The participants were overall excited about the tool and saw the potential for using it as a tool to create own mobile courses for students. This was also expressed by the participants: *"It's an easy and straight-forward way for educators who are not technically trained to use serious games"*. The majority of the participants would also use it in their own courses. However, they would also prefer to have even more options to analyze the students' interactions with the game (more advanced analytics). While the overall rating of usability of the system was very high, we also want to express the strong limitations of this study due to the small number of study participants (10). Also, participants were not required to create courses and course items, which are required for their own courses. To summarize, this study was designed as a preliminary study to gain an overview of the current state of the platform and does not replace a large-scale study. However, it already shows the potential and also the demand for such tools for creating educational game-based courses.

5 Discussion and Conclusion

The goal of the design and development of sCool was to provide a first prototype of a game-based mobile learning application, which can be used by teachers to create and adapt course material based in the own course content and use and transmit it to a playful and entertaining environment (an educational game)

DOI: https://doi.org/10.3217/978-3-85125-609-3-11

without any development or game design skills. In this paper, we have introduced the design and development of such as system and also presented a first study to evaluate both parts: first, a playful application, which is designed with game design strategies to engage learners to explore theoretical concepts and train these concepts in a practical part. Second, a web-based tool to enable teachers to add and edit learning content which feeds into the game application. The game application was designed with a focus on engaging and immersing learners into the playful learning experience. The web-based application was designed with focus on user-friendly design to engage teachers to use this tool.

We evaluated both parts in first user studies with the two different target groups (learners and teachers). In the first study with the mobile game application, we were able to show the potential of such playful applications to learn and especially engage and immerse learners. Almost all study participants think that such an application is a good supplement to regular learning and they also mentioned that they learned something new when using this application. Immersion and engagement were both rated as high. However, is was also mentioned that this game only supports a shooting-based gameplay and some thought it might be too violent. For future developments, we plan to include different sorts of gameplay, which are less violent and also attract different target groups. In the second study, we evaluated sCool for teachers with a focus on measuring usability and gain information about how users interact and learn this new software. The usability was rated with a total SUS score of 84.24 as very high and educators, which were also less trained in the use of such applications were able to use it. They especially appreciated the possibility to analyze and assess the students' interaction with the questions and the system. However, it was also mentioned that they would require more assessment and analytics features for a proper student assessment. Even though both studies were limited due to the small number of study participants, these first results demonstrate the potential of this application for both learners and teachers and also show the necessity of creating playful mobile educational tools, which are designed for both target groups in a classroom.

References

1. Obama, B. *Remarks by the President at White House Science Fair* 2015.
2. Murray, M. *To Address STEM Shortage, U.S. Employers Need Talented Immigrants* https://www.forbes.com/sites/realspin/2017/02/09/to-address-stem-shortage-u-s-employers-need-talented-immigrants/#3a51ec55c342 (2018).
3. UNESCO. *Cracking the code: Girlsfffdfffdfffd and womenfffdfffdfffds education in science, technology, engineering and mathematics (STEM)* http://unesdoc.unesco.org/images/0025/002534/253479E.pdf (2018).

DOI: https://doi.org/10.3217/978-3-85125-609-3-11

4. Gütl, C. *et al. Expectations of the generation NeXt in higher education: Learning engagement approaches in information sciences subjects* in *Interactive Collaborative Learning (ICL), 2015 International Conference on* (2015), 205–214.
5. Rieber, L. P. Seriously considering play: Designing interactive learning environments based on the blending of microworlds, simulations, and games. *Educational technology research and development* **44,** 43–58 (1996).
6. Michael, D. R. & Chen, S. L. *Serious games: Games that educate, train, and inform* (Muska & Lipman/Premier-Trade, 2005).
7. eMarketer. *Mobile Phone, Smartphone Usage Varies Globally* `https://www.zooniverse.org` (2018).
8. Wen, C. & Zhang, J. Design of a microlecture mobile learning system based on smartphone and web platforms. *IEEE Transactions on Education* **58,** 203–207 (2015).
9. Eichler, M. L., Perry, G. T., Lucchesi, I. L. & Melendez, T. T. in *Encyclopedia of Information Science and Technology, Fourth Edition* 6376–6387 (IGI Global, 2018).
10. Huizenga, J., Admiraal, W., Akkerman, S. & Dam, G. t. Mobile game-based learning in secondary education: engagement, motivation and learning in a mobile city game. *Journal of Computer Assisted Learning* **25,** 332–344 (2009).
11. Chee, K. N., Yahaya, N., Ibrahim, N. H. & Hasan, M. N. Review of mobile learning trends 2010-2015: a meta-analysis. *Journal of Educational Technology & Society* **20,** 113–126 (2017).
12. Sáez-López, J.-M., Román-González, M. & Vázquez-Cano, E. Visual programming languages integrated across the curriculum in elementary school: A two year case study using fffdfffdfffdScratchfffdfffdfffd in five schools. *Computers & Education* **97,** 129–141 (2016).
13. Slany, W. Tinkering with Pocket Code, a Scratch-like programming app for your smartphone. *Proceedings of Constructionism* (2014).
14. Ortiz, O. O., Franco, J. Á. P., Garau, P. M. A. & Martín, R. H. Innovative mobile robot method: improving the learning of programming languages in engineering degrees. *IEEE Transactions on Education* **60,** 143–148 (2017).
15. Code Hunt. *microsoft.com/en-us/research/project/code-hunt/* `https://www.microsoft.com/en-us/research/project/code-hunt/` (2018).
16. Browne, K. & Anand, C. *Gamification and serious game approaches for introductory computer science tablet software* in *Proceedings of the First International Conference on Gameful Design, Research, and Applications* (2013), 50–57.
17. Kojic, A. *Design and Implementation of an Adaptive Multidisciplinary Educational Mobile Game* MA thesis (Graz University of Technology, Austria, 2017).
18. Togelius, J., Shaker, N. & Nelson, M. J. Procedural Content Generation in Games: A Textbook and an Overview of Current Research/J. *Togelius, N. Shaker, M. NelsonfffdfffdfffdBerlin: Springer* (2014).

DOI: https://doi.org/10.3217/978-3-85125-609-3-11

19. Berto, F. & Tagliabue, J. Cellular automata (2012).
20. Kojic, M. *Procedural Generation in Multidisciplinary Educational Video Game* MA thesis (Graz University of Technology, Austria, 2017).
21. Hunicke, R. & Chapman, V. AI for Dynamic Difficulty Adjustment in Games. *Association for the Advancement of Artificial Intelligence (AAAI),* 2–5 (2004).
22. Brockmyer, J. H. *et al.* The development of the Game Engagement Questionnaire: A measure of engagement in video game-playing. *Journal of Experimental Social Psychology* **45,** 624–634 (2009).
23. Brooke, J. *et al.* SUS-A quick and dirty usability scale. *Usability evaluation in industry* **189,** 4–7 (1996).
24. Kay, R. H. & Loverock, S. Assessing emotions related to learning new software: The computer emotion scale. *Computers in Human Behavior* **24,** 1605–1623 (2008).