Supporting Learning Programming Using Educational Digital Games

Ivona Frankovic, Natasa Hoic-Bozic, Martina Holenko Dlab, Marina Ivasic-Kos
University of Rijeka, Department of Informatics, Rijeka, Croatia
ifrankovic@inf.uniri.hr
natasah@inf.uniri.hr
mholenko@inf.uniri.hr
marinai@inf.uniri.hr

Abstract: Contemporary education promotes the usage of digital games to support learning programming. Such games are not designed for entertainment, but specifically for educational purposes. Students start from a very early age to interact with computers through games and consequently, react positively on the educational digital games. When used for education, digital games have the potential to motivate students towards active participation and interaction. In addition, educational digital games make complex and abstract topics like programming more friendly.

Games that support learning programming enable students to learn programming concepts using visual interfaces and interesting environments. In these games, younger students are usually expected to move the main character around the given path and perform a series of tasks (e.g. collect objects, skip the obstacles). Instead in a textual editor, students are coding through drag and drop interactions, and create programs that include basic programming concepts like sequence, variables, loops, and conditions. Furthermore, digital games are a good for visualizing the execution of algorithms that are often hard to understand. To make the game more interesting, the main character is often presented as an animated robot, animal or a cartoon character familiar to the students.

The paper explores the possibilities of using existing digital games in learning basic programming skills, highlighting which programming concepts the game supports and for which it is recommended and appropriate to be used. This is the first step in the design of an educational game that is intended to develop within the “Coding4Girls” and "Digital games” projects. Future activities and plans for both projects are presented in the paper.

Keywords: learning programming, computational thinking, educational digital games, project Coding4Girls, project Digital games

1. Introduction

Digital games are often used in education because such learning environments can increase students’ motivation, enhance acquisition of different skills and provide chance for repeated practice and prompt feedback (Wouters and van Oostendorp, 2016). Although digital games are intended for entertainment and fun, if used for education, their primary purpose is to provide information or assist in acquiring knowledge (Michael and Chen, 2006). Therefore, the components that make the game attractive are combined with the subject matter to be transferred to the student/player.

Educational digital games can be used for acquiring programming skills since students, by playing games, learn various programming concepts and constructs (Kazimoglu *et al.*, 2012). Games for learning programming encourage good programming practices by facing the player with different challenges or visualize the operation of some algorithms (Giannakoulas and Xinogalos, 2018). They have simple interfaces where novices code using blocks and drag and drop interactions while more experienced students use syntax of a certain programming language (Kazimoglu *et al.*, 2012). Students can also learn programming by developing simple digital games (Weintrop and Wilensky, 2015), for example with visual programing languages like Scratch (*Scratch*, 2019), Snap! (*Snap!*, 2019), and Alice (*Alice*, 2017).

There are many online games that can be used for learning programming, so the choice can be made according to the concepts that can be learned but also students’ interests. Usually, girls are not fond of direct competition and prefer problem resolution (Alserri, Mat Zin and Tengku Wook, 2018). They also prefer adventure, puzzle, social (with a rewarding system), collaborative and exploration games (Hosein, 2019).

2. Digital games for learning programming

The following section gives an overview of examples of games for learning basic concepts like sequence, loops, variables, conditionals (Brennan and Resnick, 2012) as well as more advanced concepts like functions.

LightBot (*Lightbot*, 2017) is a puzzle game in which the player guides a robot around 3D maze in order to light up the blue tiles (Figure 1). The commands are represented as tiles with symbols. The player should use the minimum number of commands (optimize code) since the area for the tiles is limited. Through the game, students can learn: sequence, loops and conditions.

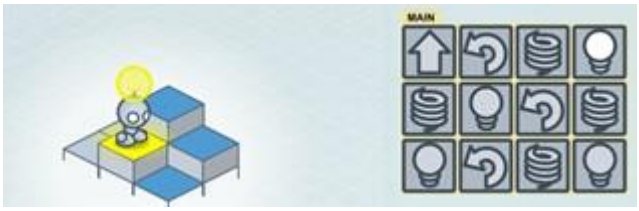

Figure 1: LightBot: Example task with solution

Dragon Dash (*Dragon Dash*, 2018) is a puzzle game in which the player should lead the dragon to the treasure (Figure 2) by putting together blocks with instructions. This game introduces instruction, sequence, and conditions.

Figure 2: Dragon blast: Example task with solution

Run Marco! (*Run Marco*, 2019) is an adventure game in which the player leads Marco or Sophia through the given path (Figure 3) to the finish, by putting together blocks with instructions. Player can learn the following programming concepts: sequence, loops, variable and conditionals.

Figure 3: Run Marco!: Example task with solution

Code Monkey (*CodeMonkey*, 2019) is a game in which player helps monkey to collect bananas (Figure 5). The player does not use blocks, but should write code using CoffeeScript. When a player faces more complex challenges, a turtle appears to help him in solving the tasks. In addition to sequences and conditions, events, operators, and functions are introduced to players.

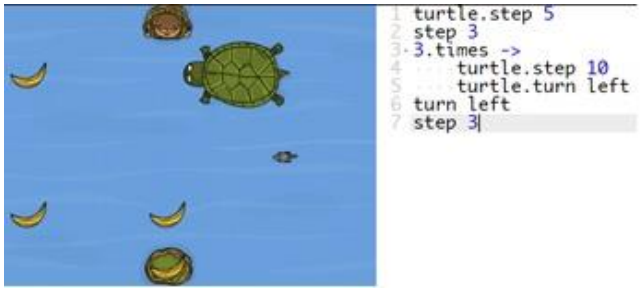

Figure 4: CodeMonkey: Example task with solution

May's Journey (*MaysJourney*, 2019) is a 3D adventure in which a girl needs to fix a collapsing world by solving secrets using a custom programming language (Figure 6). The program concepts that are taught in this game are instructions, sequence, loops, variables, conditions, and operators.

Figure 5: May's Journey: Example task with solution

CodeCombat (CodeCombat, 2019) is an adventure game where students have to write their own code (either in JavaScript or Python) to help the characters get jewels and avoid spikes and ogres (Figure 7). Through the five worlds, a lot more concepts are introduced than in the games mentioned earlier. New concepts are methods, parameters, object property, arrays, objects, and advanced techniques. The game is appropriate for students of upper grades of elementary school.

Figure 6: CodeCombat: Example task with solution

The above list of games shows that puzzle and adventure games are mostly used for introducing basic programming concepts to young students. For higher class students, 3D adventures have been used that introduce concepts of object paradigm and allow where they can write code and use numerous object programming concepts.

3. Coding4Girls and Digital Games projects

Coding4Girls is an Erasmus+ project that aims to promote the development of programming skills among girls through digital games, and Digital Games is a project funded by the University of Rijeka that will explore the possibilities of application of Game Based Learning (GBL). The main goal of the project is to increase the motivation for learning programming skills in primary school.

Target groups for projects are students between 10 and 15 years who learn the basics of programming. The GBL will be combined with the design thinking approach (Wrigley and Straker, 2017) which encourages the students to generate ideas for solving real-life problems, as well as to prototype and test their solutions.

Unlike the before mentioned games, the Coding4Girls game for learning basic programming concepts will be a 3D single player first person adventure game. The game will be developed in Unity and will include various logical games such as puzzles, mazes and other problem-solving mini-games that are chosen to be attractive to girls.

The game will be divided into chapters named "challenges" that relate to programming concepts such as loops, conditionals, etc. At the beginning of each chapter, a task related to a concept will be displayed to students. Students will play different types of mini-games (puzzle games) to reach the goal where they will need to solve the program assignments in Snap!. The goal of mini-games is to entertain the players, but also to introduce new programming concepts more easily. On the other hand, Snap! environment will help them to use and apply these concepts. Another way to attract girls is to implement additional game features after successful task solving like sharing results on social media, including photos from social media in the game environment, etc. In order to integrate a design thinking pattern into the game, at the beginning of each chapter, a real-life problem will be presented and students will discuss and share the ideas of possible solutions. Once the brainstorming phase is achieved, each of the students would start the loop of challenges and Snap! exercises necessary to solve the given problem.

To monitor the students' progress in programming and to manage game to better adapt to student skills, a web platform will be developed for teachers. The role of the teachers is to prepare and assign the elements of the games to unlock exercises/hints according to player progress.

During the Digital Games project contemporary pedagogical-technological framework for the use of the above-mentioned game for learning programming concepts in schools as well as learning scenarios will be developed. The Design Based Research (DBR) approach (Wang and Hannafin, 2005) will be used to evaluate the proposed framework while implementing the scenarios for learning and teaching programming in primary schools in Croatia. DBR will be implemented in three iterations. In the first iteration, the game will explain just one programming concept in order to familiarize students with the game environment. Subsequently, the game will be adjusted according to the obtained results and feedback and new concepts will be added to the game. In the last iteration, the entire framework will be tested.

4. Conclusions

The paper presents work in progress in the context of two projects Coding4Girls and Digital Games aiming to develop Unity based educational game for learning programming concepts and exploring the possibilities of using it for learning and teaching in primary schools.

For many students, playing digital games is the favorite activity in their free time, which is why digital games have become popular for educational purposes as well. The digital game can offer various possibilities for learning and one of them, presented in this paper, is to learn programming concepts in primary school. Applying digital games in education makes possible to avoid negative stereotypes associated with programming, especially among girls, and to overcome the gap between male and female participation in computer science education and careers.

Acknowledgment

The research has been co-funded by the Erasmus+ Programme of the European Union under the project „Coding4Girls" (2018-1-SI01-KA201-047013) and by University of Rijeka under the project "Digital games in the context of learning, teaching, and promoting inclusive education" (uniri-drustv-18-130).

References

Alice (2017). Available at: https://www.alice.org/ (Accessed: 18 April 2019).

Alserri, S., Mat Zin, N. A. and Tengku Wook, T. S. M. (2018) 'Gender-based Engagement Model for Serious Games', *International Journal on Advanced Science, Engineering and Information Technology*, 8(4), p. 1350. doi: 10.18517/ijaseit.8.4.6490.

Brennan, K. and Resnick, M. (2012) 'New frameworks for studying and assessing the development of computational thinking', *annual American Educational Research Association meeting, Vancouver, BC, Canada*, pp. 1–25. doi: 10.1.1.296.6602.

CodeCombat (2019) *CodeCombat - Learn how to code by playing a game*. Available at: http://codecombat.com/ (Accessed: 20 April 2019).
CodeMonkey (2019). Available at: https://www.playcodemonkey.com/home (Accessed: 20 April 2019).
Dragon Dash (2018). Available at: https://www.tynker.com/hour-of-code/dragon-dash (Accessed: 20 April 2019).
Giannakoulas, A. and Xinogalos, S. (2018) 'A pilot study on the effectiveness and acceptance of an educational game for teaching programming concepts to primary school students', *Education and Information Technologies*, pp. 1–24. doi: 10.1007/s10639-018-9702-x.
Hosein, A. (2019) 'Girls' video gaming behaviour and undergraduate degree selection: A secondary data analysis approach', *Computers in Human Behavior*. Elsevier, 91(September 2018), pp. 226–235. doi: 10.1016/j.chb.2018.10.001.
Kazimoglu, C. *et al.* (2012) 'A Serious Game for Developing Computational Thinking and Learning Introductory Computer Programming', *Procedia - Social and Behavioral Sciences*, 47, pp. 1991–1999. doi: 10.1016/j.sbspro.2012.06.938.
Lightbot (2017). Available at: http://lightbot.com/flash.html (Accessed: 20 April 2019).
MaysJourney (2019). Available at: https://maysjourney.com/ (Accessed: 20 April 2019).
Michael, D. and Chen, S. (2006) *Serious games: Games That Educate, Train, and Inform*. Thomson Course Technology PTR.
Run Marco (2019). Available at: https://runmarco.allcancode.com/ (Accessed: 20 April 2019).
Scratch (2019). doi: 10.1016/j.jaci.2011.12.968.
Snap! (2019). Available at: http://snap.berkeley.edu/about.html (Accessed: 18 April 2019).
Weintrop, D. and Wilensky, U. (2015) 'To beta block or not to beta block; that is the question', *Critical Care*, 19(1). doi: 10.1186/s13054-015-1059-6.
Wouters, P. and van Oostendorp, H. (2016) 'Overview of Instructional Techniques to Facilitate Learning and Motivation of Serious Games', *Instructional Techniques to Facilitate Learning and Motivation of Serious Games*, pp. 1–16. doi: 10.1007/978-3-319-39298-1_1.
Wrigley, C. and Straker, K. (2017) 'Design Thinking pedagogy: the Educational Design Ladder', *Innovations in Education and Teaching International*. Routledge, 54(4), pp. 374–385. doi: 10.1080/14703297.2015.1108214.