FunPlogs – A Serious Puzzle Mini-Game for Learning Fundamental Programming Principles Using Visual Scripting

Robin Horst, Ramtin Naraghi-Taghi-Off, Savina Diez, Tobias Uhmann, Arne Müller, and Ralf Dörner

RheinMain University of Applied Sciences, Wiesbaden, Germany
firstName.lastName@hs-rm.de
www.hs-rm.de

Abstract. Learning to program can be a tedious task for students. The intrinsic motivation towards games can help to facilitate the first steps in such learning tasks. In this paper we introduce FunPlogs – a serious puzzle mini-game for learning fundamental programming principles. We use visual scripting aspects within this game. These must be used by the students to solve spatial puzzle-like tasks. Within this game we integrate a user-driven content creation approach for the game, so that students can cooperatively create new levels. We show the feasibility of the game concept in a prototype implementation and indicate a high joy of use during a user study.

Keywords: Computer games · Content creation · Computer science education · Visual scripting language · Virtual reality.

1 Introduction

Getting students interested in nature sciences is a challenge that also applies for the sub-field of computer science. Motivating to learn programming-principles can be a tough challenge, as well, since students often have intrinsic motivation towards games but tend to have a lack of interests in learning activities [20]. However, existing research has shown that games can also be used as a beneficial medium for computer science education [17, 5, 11, 24] and especially programming education [1, 16]. Such games that have a primary focus other than entertainment are called serious games. Approaches in the educational domain do vary depending on aspects like the target audience, spatial setting, learning type, utilization time, game genre and many more. One field of application of short mini games is microlearning [7, 8], where learning content may only endure few minutes and should be used as independent self-contained learning sessions.

Another challenge in this domain is creating content (e.g. different levels) for such serious games [12]. Providing predefined levels created by game designers increases development effort and results in a finite set of learning content. Integrating user-generated content for serious learning games can help to decrease such development effort and increase the reusability of the software [15].

Specifically letting users create content within the game itself is mentioned to be promising and engaging, for example letting users create their own story [22].

In this paper we introduce FunPlogs – a serious puzzle mini-game for learning fundamental programming principles using visual programming. With this work we make the following contribution:

1. We propose a serious game concept to give students an understanding of basic principles of modern procedural programming, like loops and if-then-else decisions. The core of the game for the students is to solve puzzle tasks. They have to use visual scripting to express spatial moving tasks.
2. A user-driven level creation method is provided within this game concept. Content is created collaboratively and asymmetrically by the students themselves. The creation process involves virtual reality (VR) methodology.
3. With a prototype and a user study we show the feasibility of the concepts and indicate that students have fun by using our learning application.

2 Related Work

We relate our work to existing work on computer science education games and user-driven content creation.

Pioneering and famous work by Resnick et al. [21] introduce the visual programming language Scratch. It indicates the success of visual approaches in programming education. It is not directly a learning game, but enables laymen to program and create their own games by providing fundamental programming aspects as visual components that can be connected. Predecessors of Scratch and further pioneering work, such as Squeak [3, 10], Etoys[1] and Alice [18, 6] provide further insights in the matter of visual programming, VR and learning purposes. These works provide a basis for several follow-up projects (e.g. [12, 19]).

Mildner et al. [15] propose a system for the creation of custom-made serious games with user-generated learning content. They enable teacher to create mini games for their students. Web-based authoring is used to create content for their Knowledge Defence game, which lies in the genre of tower defence.

QuizPACK [4], by Brusilovsky and Sosnosovsky uses quizzes to communicate knowledge about programming and follows a similar idea as Mildner et al.. Both use a dedicated authoring application and a separate game client application similar as described in the paper on Word Domination [14]. Word Domination draws from first person shooters and quiz elements concerning its game genre.

Work by Johnson and Valente [9] and by Mehm et al. [13] specifically addresses collaborative authoring paradigms for serious games. They propose such concepts for different application areas, which are games for language and culture and health respectively. Johnson and Valente differentiate between two types of author groups which are domain experts and game experts, such as artists, designers and programmers. The relation between these groups is that the game experts provide authoring tools for the domain experts in consultation with

[1] www.squeakland.org (Accessed 22.08.2019)

them. Mehm et al., however, propose a similar relation, but subdivides the game experts group into the specific roles in a game creation cycle, which are game designers, game programmers, authoring tool programmers and expert authoring tool users. They describe a fixed process in which these groups work together to create serious games for the field of health education. Masuch and Rueger [12] investigate further challenges in collaborative game design and developing learning environments for creating games. Based on their experience with Squeak [10, 3] they perform an analysis of game design processes with OpenCroquet [23] as a platform for collaborative design for online games.

Torrente et al. [26] and Yessad et al. [27] propose instructor-oriented authoring tools for educational video games. They do not draw conclusions on collaborative aspects, but propose systems that help domain experts author entire games in visual editors. Barron-Estrada et al. [2] build on similar concepts as Torrente et al. and Yessad et al. and propose "CodeTraining", an authoring tool for creating content for a gamified programming environment. They enable authors to define specific programming exercises for students. Students can earn points and rise in a leader board visible by other students.

All games and authoring procedures mentioned above provide a static game scenario and genre and let expert-users create content for this scenario, whereas Pex4Fun [25] serves several genres in the form of mini games. One is a puzzle game that takes parameters and return values and let users choose appropriate arguments. The authors of Pex4Fun, however, do not provide direct authoring possibilities, but provide suggestions on how to integrate the puzzles and other mini games in classes. No work was found where learners create their own content for the field of computer science education within the actual game, or in a gamified process – mostly dedicated authoring tools are provided to facilitate the game and content creation.

3 FunPlogs Game Concept

FunPlogs is a serious puzzle and building mini game that makes use of user-created levels to learn fundamentals of programming, as concepts of while-loops and if-then-else decisions. The game concept of FunPlogs is divided into two playable scenes: 1) One for creating playable content together with other students – the *building scene*. And 2) one for actually playing the learning game – the *scripting scene*.

The game flow is build up by iteratively switching between the building scene and the scripting scene. However, the building scene and the scripting scene can be use several times, so that a reflexive and symmetric game flow graph results (Fig. 1) Both scenes are described in detail in the following sections.

3.1 Scripting Scene

The scripting scene is the core of the learning application. It represents a casual puzzler mini game. Users must guide the player character from an initial starting

Fig. 1. Game flow of FunPlogs. Building new levels and playing the learning game is illustrated as an iterative process. Built level can also be saved for later usage, so that playing or building can also be repeated.

-

point to a specific goal position. For this the character must find a way on which he does only use predefined blocks for his path. He must not drop down from the blocks to the bottom of the play area. An example level can be seen in Fig. 2.

Users can direct the virtual character along the path by executing an instruction (Fig. 2 instruction panel) that is made up from several scripting commands (Fig. 2 scripting command menu). The commands in the instruction panel are processed sequential.

A level for FunPlogs is formed by building blocks of different colors. The colors have no impact on the game logic, but contribute to the aesthetic of the game and establish clues for users to orient. The blocks are aligned on a grid, and they can only be stacked upwards or placed on the ground-plane as the lowest level of the grid.

We provide basic movement tasks for solving the spatial task of navigating the character on these blocks: 1) Move forward, 2) move upward, 3) move downward, 4) turn right and 5) turn left. All levels that are designed with the building blocks and the constraints mentioned above can be solved with these basic commands.

Both complex and easy levels can be solved with just using a longer sequence of the basic commands. But users can also use if-/while-statements to see how the instructions can be simplified and how the command count can be reduced. This illustrates to them the use and benefits of if-/while-statements. Whether levels are rather complex or easy is decided by the actual users that create the levels themselves within the building scene.

3.2 Building Scene

The building scene is a sandbox mode of the play area from the scripting scene. The basic principle for creating levels for FunPlogs contrary to the scripting scene involves two players – a *desktop-player* and a *VR-player*. The desktop-player interacts with the system similar as in the scripting scene through a desktop-PC. Players can throw building blocks onto the ground-plane within his top-down view (Fig. 3) and sees the point of view (POV) of the VR-player on his interface in the top left corner.

The VR-player is located on the initially empty ground-plane. These players see the play area from a first person point of view (Fig. 4). They place the dropped down building blocks on the grid to form the foundation for a walkable

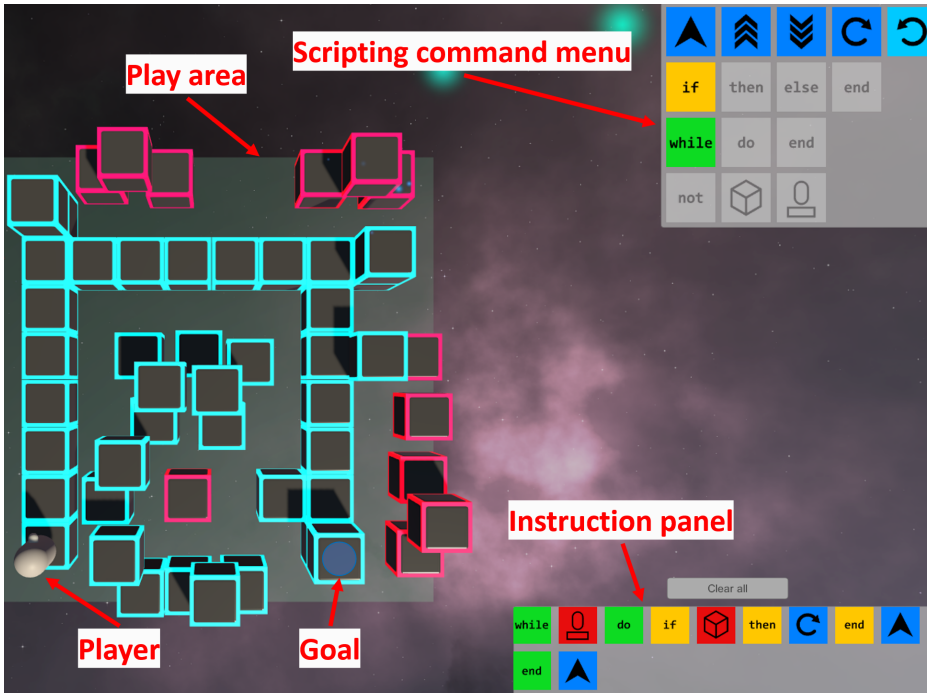

Fig. 2. The top-down point of view of the desktop player during the scripting mode. The interface contains the play area, the scripting command menu and the instruction panel. We furthermore show the virtual player as the starting point and the goal of the puzzle.

-

path for the virtual character. The VR-player furthermore chooses a starting point for the virtual character and the goal.

Both player-types can suggest certain positions for a drop down or block placement by positioning a marker cube ((Fig. 4 purple cube top left)) which then hovers over a specific rectangle of the grid. By collaborating and interacting with each other the level is finished. This collaboration within the virtual environment is of asymmetric nature, since one player uses immersive VR technology and the desktop player uses abstracted desktop interactions.

3.3 Prototype Implementation

For our prototype implementation we used the Unity3D game engine as a foundation. The graphical user interface, like the instruction panel of the scripting command menu (Fig. 2, was realized using basic Unity UI-elements. Remaining game objects like the ground-plane, the blocks and the player character were made from Unity primitive objects and textures.

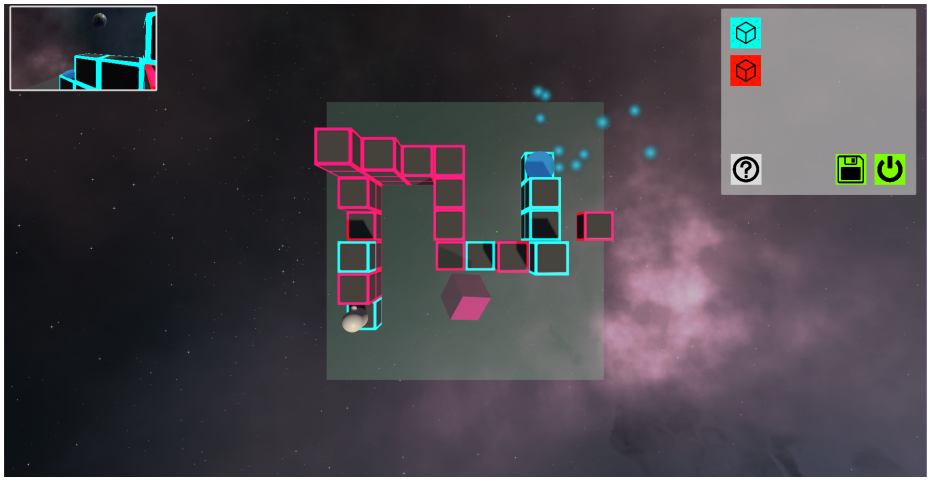

Fig. 3. The view of the desktop-player during building mode.
-

The integration of the asymmetric aspect of the virtual environment was more challenging. We explored two possibilities for implementing the multi-player building scene that uses both VR and desktop hardware and interactions. 1) The game can have been developed as a common distributed multi-player game. Two distinct applications are being synchronized, each handling one player class and controlling one specific set of hardware. 2) We decided to implement both classes in one game application that runs the whole game logic and controls the VR and the desktop part. This eliminates the need of a synchronization and facilitates implementations of asymmetric interaction features, like providing the first person view of the VR-player to the desktop-player (Fig. 3 top left). To implement these and all other VR-relating features we used the Virtual Reality Toolkit (VRTK)[2]. We used an HTC Vive VR setup.

4 Evaluation

We conducted an initial user study for evaluating the game concept and draw conclusions about the proposed learning effect of the visual scripting aspects and the user-driven content creation method. This study involved 8 participants (5 male; from 15 to 53 years with Ø 29,6 years). Half of them stated to have little prior experience in programming.

The study started with a user test. Two users tested simultaneously. The test scenario is illustrated in Fig. 5. Participants first tested an example level in the scripting mode to make familiar with the game. After some minutes of free testing, the two participants joined in a game. They were asked to build a level

[2] https://vrtoolkit.readme.io/ (Accessed 03.04.2019)

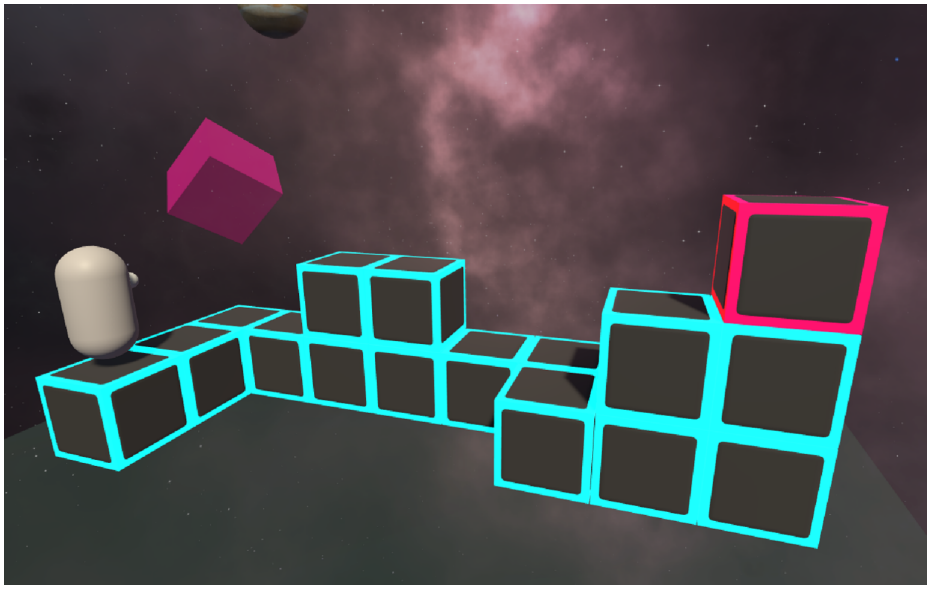

Fig. 4. The view on a FunPlogs level through the eyes of the virtual reality player within the building scene of the game.

-

in the building scene together. The participants were asked to switch the player roles (VR/desktop) during this step and then could use their own levels within the scripting scene.

After further testing, the participants were asked to fill out an anonymized questionnaire. This questionnaire consisted of demographic questions, followed by 4 questions on the perceived joy of use, the comprehensibility and the scripting tasks. These questions were translated into German as the participants were native Germans: (Q1) How much fun did you have during the creation of the game scenes at the desktop PC? (2) How much fun did you have during the creation of the game scenes using the VR? (3) Do you feel confident that you have understood how an "if"-clause works? (4) Do you feel confident that you have understood how an "while"-loop works? A 5-point likert-scale was used to capture the data. There was also space for free text comments. The answers are illustrated in Fig. 6.

4.1 Analysis of the User Study Results

With an average of Ø 5.0 points and a SD of 0.0 participants stated to have fun in both roles during the collaborative and asymmetric creation of the FunPlogs levels. With Ø 4.85 and SD 0.34 they stated to have understood how if-clauses and with Ø 4.83 and SD 0.37 how while-loops do work. Only considering par-

Fig. 5. The valuation took place in a loose but controlled laboratory environment. Left: VR-player. Right: Desktop-player

-

ticipants who explicitly had no programming experience, the values were Ø 4.66 and SD 0.57 for both questions.

In observations during the tests and in free text comments participants stated that they specifically found the collaborative building scene very playful. They furthermore expressed they like the very simple principle of the visual scripting game. Participants also stated that the collaboration during the content-authoring did "randomize" the design of the levels. One participant explicitly mentioned a lack of the VR-player during the actual scripting mode.

4.2 Discussion of the Results

The study generally indicates that participants perceived a high joy of use while playing FunPlogs. Furthermore, it is indicated that despite the simple game concept, complex matters as the while-loop or an if-clause could be transported to programming laymen. Participants needed only few minutes to complete a level of FunPlogs so that an application in the educational field of microlearning could be beneficial for learners.

It can be seen as an advantage that participants found that the collaborative user-driven content creation did veil the level design. The levels therefore were not too clear or predictable during the scripting mode even though they were self-created minutes before the actual usage.

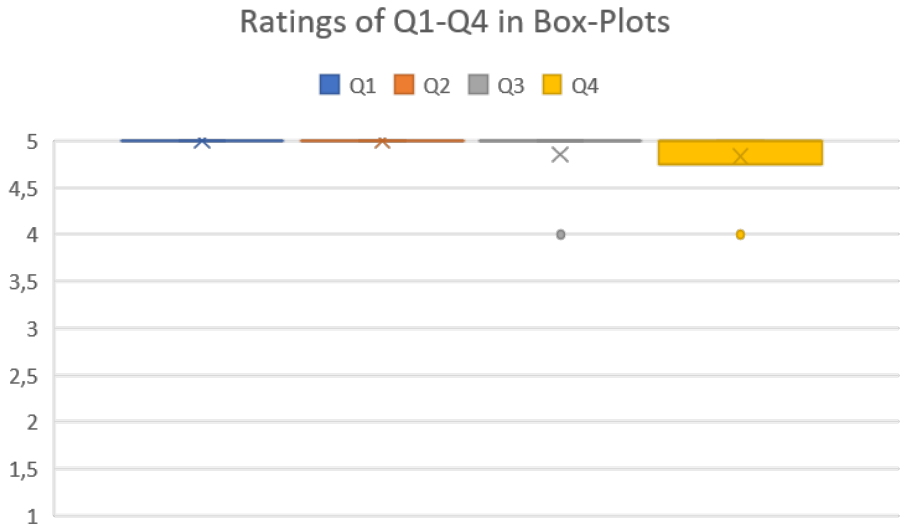

Fig. 6. Box-plot diagrams that illustrate the outcome of the 4 questions. High values represent a positive connotation.

-

5 Conclusions and Future Work

In this paper we have proposed FunPlogs – a serious puzzle mini-game for learning fundamental programming principles using visual scripting. We have described the concept of the learning game and proposed a collaborative and asymmetric way of user-driven level creation for the game. We could indicate a high joy of use during the usage of FunPlogs on the one hand and exemplary creating a basic understanding of while-loops and if-clauses on the other hand.

In future work, we will investigate in two major aspects. 1) We will maintain more programming aspects in the scripting mode, so that future implementations can transport more knowledge to students. 2) The collaborative VR-aspect of the game will be increased since participants mentioned a high level of joy during collaborating in the building scene. Integrating a VR-player within the actual scripting scene could therefore be beneficial to communicate knowledge within the serious learning game.

Acknowledgements

The work is supported by the Federal Ministry of Education and Research of Germany in the project Innovative Hochschule (funding number: 03IHS071).

References

1. Adams, J.C., Webster, A.R.: What do students learn about programming from game, music video, and storytelling projects? In: Proceedings of the 43rd ACM technical symposium on Computer Science Education. pp. 643–648. ACM (2012)
2. Barrón-Estrada, M.L., Zatarain-Cabada, R., Lindor-Valdez, M.: Codetraining: An authoring tool for a gamified programming learning environment. In: Mexican International Conference on Artificial Intelligence. pp. 501–512. Springer (2016)
3. Bouras, C.J., Poulopoulos, V., Tsogkas, V.: Squeak etoys: Interactive and collaborative learning environments. In: Handbook of research on social interaction technologies and collaboration software: Concepts and trends, pp. 417–427. IGI Global (2010)
4. Brusilovsky, P., Sosnovsky, S.: Individualized exercises for self-assessment of programming knowledge: An evaluation of quizpack. Journal on Educational Resources in Computing (JERIC) **5**(3), 6 (2005)
5. Chandel, P., Dutta, D., Tekta, P., Dutta, K., Gupta, V.: Digital game based learning in computer science education. CPUH Res. J **1**(2), 33–37 (2015)
6. Cooper, S., Dann, W., Pausch, R.: Alice: a 3-d tool for introductory programming concepts. Journal of Computing Sciences in Colleges **15**(5), 107–116 (2000)
7. Hug, T.: Micro learning and narration: exploring possibilities of utilization of narrations and storytelling for the design of micro units and didactical micro-learning arrangements. Proceedings of Media in Transition (2005)
8. Hug, T.: Microlearning: a new pedagogical challenge (introductory note) (2005)
9. Johnson, W.L., Valente, A.: Collaborative authoring of serious games for language and culture. In: Proceedings of SimTecT. vol. 2008 (2008)
10. Kay, A.C.: Computers, networks and education. Scientific American **265**(3), 138–149 (1991)
11. Liu, E.Z.F., Chen, P.K.: The effect of game-based learning on students learning performance in science learning–a case of conveyance go. Procedia-Social and Behavioral Sciences **103**, 1044–1051 (2013)
12. Masuch, M., Rueger, M.: Challenges in collaborative game design developing learning environments for creating games. In: Third International Conference on Creating, Connecting and Collaborating through Computing (C5'05). pp. 67–74. IEEE (2005)
13. Mehm, F., Hardy, S., Göbel, S., Steinmetz, R.: Collaborative authoring of serious games for health. In: Proceedings of the 19th ACM international conference on Multimedia. pp. 807–808. ACM (2011)
14. Mildner, P., Campbell, C., Effelsberg, W.: Word domination. In: International Conference on Serious Games. pp. 59–70. Springer (2014)
15. Mildner, P., John, B., Moch, A., Effelsberg, W.: Creation of custom-made serious games with user-generated learning content. In: Proceedings of the 13th Annual Workshop on Network and Systems Support for Games. p. 17. IEEE Press (2014)
16. Moreno, J.: Digital competition game to improve programming skills. Journal of Educational Technology & Society **15**(3), 288–297 (2012)
17. Papastergiou, M.: Digital game-based learning in high school computer science education: Impact on educational effectiveness and student motivation. Computers & education **52**(1), 1–12 (2009)
18. Pausch, R., Burnette, T., Capeheart, A., Conway, M., Cosgrove, D., DeLine, R., Durbin, J., Gossweiler, R., Koga, S., White, J.: Alice: Rapid prototyping system for virtual reality. IEEE Computer Graphics and Applications **15**(3), 8–11 (1995)

19. Pinna, S., Mauri, S., Lorrai, P., Marchesi, M., Serra, N.: Xpswiki: An agile tool supporting the planning game. In: International Conference on Extreme Programming and Agile Processes in Software Engineering. pp. 104–113. Springer (2003)
20. Prensky, M.: Digital game-based learning. Computers in Entertainment (CIE) **1**(1), 21–21 (2003)
21. Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J.S., Silverman, B., et al.: Scratch: Programming for all. Commun. Acm **52**(11), 60–67 (2009)
22. Robertson, J., Good, J.: Story creation in virtual game worlds. Communications of the ACM **48**(1), 61–65 (2005)
23. Smith, D.A., Kay, A., Raab, A., Reed, D.P.: Croquet-a collaboration system architecture. In: First Conference on Creating, Connecting and Collaborating Through Computing, 2003. C5 2003. Proceedings. pp. 2–9. IEEE (2003)
24. Steiner, B., Kaplan, N., Moulthrop, S.: When play works: Turning game-playing into learning. In: Proceedings of the 2006 conference on Interaction design and children. pp. 137–140. ACM (2006)
25. Tillmann, N., De Halleux, J., Xie, T., Bishop, J.: Pex4fun: Teaching and learning computer science via social gaming. In: 2012 IEEE 25th Conference on Software Engineering Education and Training. pp. 90–91. IEEE (2012)
26. Torrente, J., Moreno-Ger, P., Fernández-Manjón, B., Sierra, J.L.: Instructor-oriented authoring tools for educational videogames. In: 2008 Eighth IEEE International Conference on Advanced Learning Technologies. pp. 516–518. IEEE (2008)
27. Yessad, A., Labat, J.M., Kermorvant, F.: Segae: A serious game authoring environment. In: 2010 10th IEEE International Conference on Advanced Learning Technologies. pp. 538–540. IEEE (2010)