

Studying Students' Attitudes on Using Examples of Game Source Code for Learning Programming

Aristea THEODORAKI, Stelios XINOGALOS

*Department of Applied Informatics, School of Information Sciences, University of Macedonia
Egnatia 156 str., 54006 Thessaloniki, Greece
e-mail: tm1043@uom.edu.gr; stelios@uom.edu.gr*

Received: March 2014

Abstract. Games for learning are currently used in several disciplines for motivating students and enhancing their learning experience. This new approach of technology-enhanced learning has attracted researchers' and instructors' attention in the area of programming that is one of the most cognitively demanding fields in Computer Science. Several educational, or else serious, games for learning programming have been developed and the first results of their evaluation as a means of learning are quite positive. In this paper, we propose using arcade games as a means for learning programming. Based on this approach students first play a simple game, such as Snake or Tetris, study its code and then extend it. In a pilot study carried out in the context of an undergraduate programming course, students studied the source code of the well-known game Snake and extended it with new functionalities. The analysis of students' replies in a questionnaire showed that using arcade games as a means of learning programming concepts enhances students' motivation for learning programming, supports them in comprehending complex concepts and engages them in carrying out programming activities.

Keywords: serious games, arcade games, programming, motivation.

1. Introduction

Game-based learning is a relatively new field in technology enhanced learning, which has gathered the interest of researchers and instructors of various subjects. As it was expected, this new approach was adopted to enhance the teaching and learning of programming, which is undoubtedly accompanied by a variety of difficulties and is characterized as a cognitively complex process that discourages several students. The combination of educational games and programming aims to change this belief by giving students strong incentives for practicing with programming, providing them a more pleasant and creative environment (Kazimoglu *et al.*, 2012) that provokes the sense of control and develops students' curiosity and imagination.

Programming microworlds and educational programming environments can be considered as the predecessors of educational, or else, serious games for programming

(Malliarakis *et al.*, 2012). The main goal of the programming microworlds is to deal with students' learning difficulties through the usage of a physical metaphor that students are familiar with, a simple programming language with a limited instruction set and simple syntax, as well as exercises that rise students' interest about programming (Brusilovsky *et al.*, 1997).

However, the educational or serious purpose games that were developed for teaching programming in recent years incorporate more attractive scenarios and graphics in comparison to programming microworlds and educational programming environments. These games are divided into two main categories (Malliarakis *et al.*, 2013): the educational games aiming to teach a specific programming unit and the holistic educational games aiming to teach all basic programming concepts. The first assessment results about game-based learning of programming are very encouraging (Kazimoglu *et al.*, 2012; Liu *et al.*, 2011).

An alternative approach that applies some elements of game-based learning of programming is the utilization of arcade games that were developed in the first place just for entertainment purposes. Based on this approach, learning programming concepts is accomplished through playing, studying and then extending the source code of such games. Representative examples of this kind of games are Snake, Tetris and PacMan. The lack of students' motivation for learning programming experienced in the programming courses of the Department of Technology Management that was recently merged with the Department of Applied Informatics, led us in the decision to carry out a pilot study using arcade games for teaching and learning programming.

The aim of the study presented in this paper is to investigate students' attitudes on using examples of arcade game source code for supporting the teaching and learning of programming. More specifically, we wanted to investigate whether such examples have a positive impact on students' motivation to learn programming. For this purpose we designed and applied a didactical intervention in the context of an undergraduate programming course and collected data for analysis using a questionnaire filled in by students. The course uses Java as a programming language and so games implemented in this programming language were examined. However, it is obvious that the same approach of using examples of arcade game source code for learning programming can be applied to other conventional programming languages as well. The rest of the paper is organized as follows. Section 2 presents briefly the main characteristics of the educational games that have been developed for teaching and learning programming, as well as the main characteristics of Java arcade games that can be used for the same purpose. Section 3 presents the research questions and the methodology of the study, while section 4 analyzes its results. Finally, in section 5 conclusions are drawn and proposals for further research are presented.

2. Educational Games and Arcade Games

Educational games

Educational games use a graphical interface and interesting scenarios similar to the ones used in digital entertainment games in order to motivate students. Some of the edu-

educational games for programming are focused on specific programming concepts. The exercises solved in their context are usually related to completing code segments, matching, and gap filling and not developing complete programs from scratch. Representative examples of educational games that focus on teaching specific programming concepts are Catacombs (Barnes *et al.*, 2008), Saving Princess Sera (Barnes *et al.*, 2008) and Prog & Play (Muratet *et al.*, 2011). However, there are also educational games that teach all the basic programming concepts/structures. These games, usually, include many levels, with each level presenting different programming concepts/structures. The player/learner must show that he or she has comprehended the concepts of each level in order to qualify to pass to the next level. Representative examples of such games are Robocode (O'Kelly and Gibson, 2006) and M.U.P.P.E.T.S. (Phelps *et al.*, 2003). A review of both types of educational games for programming is provided in (Malliarakis *et al.*, 2013).

Arcade games based on Java

In recent years the object-oriented design and software development technique has prevailed, while one of the most popular object-oriented languages is Java. Thus, several 2D games were developed using Java's Swing package of classes for creating the underlying graphical user interface (GUI) and the Java 2D application programming interface (API) for drawing two-dimensional graphics. Typical examples of such games are Snake, Tetris, Pacman, Minesweeper, Space invaders and Sokoban. Although these games were developed for entertainment purposes they use numerous programming concepts and they can be used as the basis for learning programming as well.

In order to further support novice programmers, teachers have developed specially designed classes to reduce the complexity of the aforementioned Java's tools used for implementing Java 2D games. An example is the effort of Professor Viera Proulx that has developed a related set of libraries. These libraries provide the possibility of designing the behavior of an interactive two-dimensional game, without requiring from the student to get involved with details about the design of the graphical user interface, drawing graphics and handling events (<http://www.ccs.neu.edu/javalib/World/>). The student is primarily concerned with the model of the game, while the design of the games' screens can be composed with simple geometric shapes and images.

A recent evolution of the aforementioned two approaches was the development of networked game engines that provide additional functionality to students. Such engines include integrated graphics environments in order to develop games and provide the possibility to use the relevant libraries for implementing, debugging and sharing games via internet. A typical example of an online game engine is the Fang Engine (Freely Available Networked Game Engine), which was developed with the aim of supporting undergraduate introductory programming courses based on Java. At the website of this engine (http://www.fangengine.org/index.php/Main_Page) relevant educational material is available, such as games code, tutorials, exercises, students' projects, information about courses that utilize the online game engine, as well as the source code of games that are analyzed in the corresponding textbook (Ladd and Jenkins, 2011).

It is clear that educational games developed for learning programming and arcade games that are used as a means of learning programming share some common characteristics, but they have important differences as well. In educational games for program-

ming the goal is to motivate students to carry out programming activities and learn by engaging them in playing a game with a scenario closely related to games played for entertainment purposes. In arcade games students also play the game, but they do not learn in the context of playing. Students study and extend the source code implementing the game per se, which means that the proposed activities modify the game and are not just carried out in the context of the game. Furthermore, students must have some experience in programming in order to use and extend such a game. Students in educational games learn by playing, while in arcade games students learn by playing, studying and modifying the source code of the game.

3. The Study

In this section the research questions of the study, the methodology, the data collected and analyzed for answering these questions and information about the participants are presented.

3.1. *Research Questions*

The study presented in this paper aimed at investigating the following questions:

- **Question 1:** Is students' interest and motivation in learning programming increased by the use of arcade games?
- **Question 2:** Does the use of arcade games supports students in comprehending programming concepts and acquiring problem solving capabilities?
- **Question 3:** What are students' perceptions and expectations about using arcade games for learning programming?

3.2. *Methodology*

The study presented in this paper was conducted during the academic year 2012–2013 in the Department of Technology Management at the University of Macedonia in Greece within the elective course “Network-Centric Software”, which is taught during the eighth semester. This course aims at familiarizing students with various aspects of the network-centric programming language Java for developing desktop applications, applets and web applications based on servlets and Java Server Pages. The programme of studies includes two additional programming courses in previous semesters: “Computer Programming” based on the imperative programming language C at the second semester and “Object-Oriented Design and Programming” based on Java at the third semester (Xinogalos, 2012). Therefore the students who participated in the study had already had basic knowledge of object oriented programming and the programming language Java.

A twenty-minute presentation of the Snake game was conducted during the pre-final lecture of the semester (the 12th lecture). The presentation included a brief demonstration of the game and an analysis of the most basic parts of its source code that included both known and new concepts. The purpose of the presentation was to summarize some GUI

programming concepts which were taught during the semester and also to give students a realistic example in order to understand how these concepts are implemented in the context of a game.

Students were asked to expand the aforementioned 2D game – the Snake – that is familiar to everybody, and then answer a questionnaire. The presentation and the code of the original version of the game, as well as the exercise and the questionnaire were available via the asynchronous learning platform of the University.

3.3. Data Collection

Assignment

The project assigned as homework involved expanding and modifying the Snake game (Fig. 1). In the initial version of the game the Snake has to eat as many apples as possible, avoiding the walls and its body. The body of the Snake grows up each time an apple is eaten (<http://zetcode.com/tutorials/javagamestutorial/snake/>). The most important extensions that students were asked to implement are the following: add objects in addition to the apple which can be eaten by the Snake; calculate and display the player's score when the game is over– each object gives different points; change the game's background; and add buttons (play, pause, repeat) at the bottom of the frame (Fig. 2). Furthermore, they were asked to make proposals for additional actions/functions that could contribute to the improvement of the game.

Questionnaire

Carrying out the homework based on the Snake game was a prerequisite for the completion of the questionnaire. The questionnaire included, mainly, closed type questions (Likert-scale) which were divided into four categories:

1. Questions that examine whether students practice *programming* inside and outside the university and what means of learning they prefer to use.
2. Questions about their involvement with digital *entertainment games*.

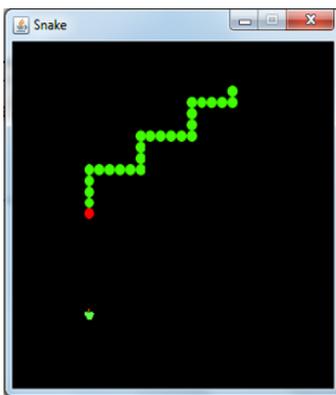


Fig. 1. The initial version of the Snake game.



Fig. 2. The extended version of the Snake game.

3. Questions about their experience in solving programming exercises based on the game and their views on *game-based-learning of programming*, as applied in this pilot study and not with its formal meaning. The questions on this category were heavily based on the questionnaire of a similar study regarding students' perceptions on using educational games for learning programming (Ibrahim *et al.*, 2011).
4. Questions for investigating students' preferences in arcade games and recording potential proposals for improvements in the game used in the pilot study. Such data was considered important for future use.

Students' replies in the questionnaire were analyzed with SPSS 17.0.

3.4. Participants

Twenty-three (23) students participated in this study of which 16 (69.6%) were men and 7 (30.4%) were women. Out of the 23 students that participated in the research only 7 attended the lecture where the game was presented. However, the material of the presentation of the Snake game, the project and the questionnaire were posted on the asynchronous learning platform that is used for the course.

4. Analysis of the Questionnaire Replies

In this section, students' replies in the questionnaire are analyzed giving emphasis on the part of the questionnaire that investigates students' perceptions on using arcade games for learning programming. Such games are characterized in the following paragraphs as simple games, since this is how students usually characterize such games in contrast to more advanced 3D entertainment games.

As mentioned the questionnaire consisted mainly of closed-type, Likert-scale questions, where the possible answers were: 1 – Strongly Disagree, 2 – Disagree, 3 – Not Sure, 4 – Agree and 5 – Strongly Agree. In Tables 1, 2 and 3 the percentage of replies falling in each one of these 5 responses, as well as the mean value (in the scale of 1 to 5) of students' replies is presented.

4.1. Interest in Programming

Hours for study per week

Almost half the students (47.8%) spend three to five hours of study weekly for each programming course taught in the Department, while fewer than one out of ten students (8.7%) spend more than five hours in programming weekly. Finally, 39.1% of the students study one to two hours per week.

Interest in programming

In order to assess the general interest of students about programming the questionnaire was enriched with questions about their personal involvement with programming out-

side the university. 30.4% (7 out of 23 students) of the population practice programming outside the university as well, with 13% (3 out of 23 students) of them spending on programming one to two hours of their free-time per week and 8.7% (2 out of 23 students) spending three to five hours weekly. On the other hand the majority of students and specifically 69.6% (16 out of 23 students) do not practice programming outside the university.

Means for learning programming

The main means for learning that are used by students in order to support and enrich their knowledge on programming concepts taught in the context of their courses – except textbooks – are video tutorials posted on YouTube and webpages that contain implemented code examples and explanations.

4.2. Interest in Digital Games

Another important aim of this study was to examine at what percentage and with what frequency students play digital entertainment games. 30.4% of the respondents play computer games, with 26.1% of them playing daily and 47.8% more than 4 times a week. In conclusion, one out of three students who participated in this survey is attracted by computer games and spends part of his/her free time in playing games. This percentage of students playing games is smaller than expected, but we have to acknowledge that the number of participants was also small.

4.3. Perceptions on Using Games for Learning Programming

Students' motivation in learning programming

Table 1 presents students' answers regarding their motivation in using games for learning. Considering students' answers the vast majority (87% – [T1.1]: 60.9% agree and 26.1% strongly agree) support the view that game-based learning projects offer several advantages, while none of them disagrees. Also, the majority of students (65.2% – [T1.2]: 34.8% agree and 30.4% strongly agree) declare that would prefer to learn programming concepts through the analysis of 2D games. Students' attitude is also positive in terms of their interest in using computer games as a means of learning programming concepts in the future (69.6% – [T1.3]: 52.2% agree and 17.4% strongly agree). Furthermore, many students prefer to solve programming exercises which are based on 2D simple games (almost 65.2% – [T1.5]), while a significant proportion of them (30.4%) are not sure about this option. Nearly all students (95.7% – [T1.6]) believe that games make programming more interesting, while nearly 3 out of 4 students (73.9% – [T1.7]) believe that if they were taught programming with games they would perform better in relevant courses.

Thus, summarizing the above results we conclude that even simple 2D games, or else arcade games, can motivate novice programmers. Most of the respondents characterize it as a very interesting approach, because it combines gaming fun and acquisition of knowledge. Utilizing games not only enhances students' interest, but also helps them to

Table 1
Replies regarding the motivation provided by games in learning programming

Statement	1-Strongly Disagree (%)	2-Disagree (%)	3-Not Sure (%)	4-Agree (%)	5-Strongly Agree (%)	Mean	Standard deviation
[T1.1] Game-based projects (e.g. Snake) give me a lots of benefits	0	0	13	60.9	26.1	4.13	.63
[T1.2] I would prefer to learn programming concepts through analysis of simple 2D games	0	8.7	26.1	34.8	30.4	3.87	.97
[T1.3] I am interested in using simple games as a means of learning programming concepts in the future	8.7	4.3	17.4	52.2	17.4	3.65	1.11
[T1.4] I am interested in using simple games as a means of learning other subjects in the future	0	0	26.1	47.8	26.1	4.0	.74
[T1.5] I prefer to solve programming exercises which are based on simple 2D games	0	4.3	30.4	34.8	30.4	3.91	.90
[T1.6] The usage of computer games makes this subject more interesting	0	0	4.3	34.8	60.9	4.57	.59
[T1.7] If I was taught programming with simple games I would perform better in relevant courses	0	8.7	17.4	56.5	17.4	3.83	.83

easier comprehend difficult concepts as it pronounces the ability to realize in real-time practical matters of programming concepts. This fact combined with students' belief that this alternative way of learning programming can help them improve their performance in relevant courses results in an increased interest in using this approach in the future. Moreover, it is worth noting that these results were recorded for a sample of students for which only one out of three students plays digital entertainment games.

Students' cognitive development

Table 2 presents students' answers regarding the contribution of games to their cognitive development. More than half the students ([T2.1]: 65.2%) support that games like Snake help them develop their critical thinking, while the rest of them are just not sure for this. Moreover, students' replies show that nearly three quarters of them (73.9% – [T2.2]: 39.1% agree and 34.8% strongly agree) consider solving exercises in the context of games, such as Snake, very interesting. Looking for answers in the context of game-based exercises is characterized by the majority of students as an encouraging activity ([T2.4]: 39.1% agree and 34.8% strongly agree). Another important finding is that students state that the game strengthened their efforts in comprehending the programming concepts implemented in it ([T2.5]: 30.4% agree and 34.8% strongly agree). So, it is not surprising that more than 80% ([T2.3]) of the respondents agree or strongly agree that it is worth trying using games for learning new programming concepts in the future and only 13% disagree.

Thus, it seems that students consider that games support them in developing an algorithmic way of thinking. Learners find problem solving in the context of game-based exercises both very interesting and encouraging, and thus their efforts are strengthened

Table 2
Replies regarding the contribution of games at Cognitive Development

Statement	1-Strongly Disagree (%)	2-Disagree (%)	3-Not Sure (%)	4-Agree (%)	5-Strongly Agree (%)	Mean	Standard deviation
[T2.1] Games like this help me develop critical thinking	0	0	34.8	60.9	4.3	3.7	.56
[T2.2] Solving problems in the context of simple games (e.g. Snake) is very interesting	0	4.3	21.7	39.1	34.8	4.04	0.88
[T2.3] It is worth to try using games for learning new programming concepts in the future	0	13	4.3	39.1	43.5	4.13	1.01
[T2.4] Looking for answers about exercises based on games is an encouraging activity	0	0	26.1	39.1	34.8	4.09	.79
[T2.5] This game strengthened my efforts for comprehending the programming concepts implemented in it	0	13	21.7	30.4	34.8	3.87	1.06

towards comprehending programming concepts and accomplishing their tasks. Future use of games for learning new programming concepts is anticipated from the majority of students.

Students' expectations

Table 3 presents students' answers regarding their expectations about the utilization of the game-based learning approach in programming courses and other Computer Science subjects. This approach seems to have a positive impact on the majority of students. More specifically the vast majority of students ([T3.1]: 91.3%) would like to have more opportunities to learn using the game-based learning approach. More than half the re-

Table 3
Students' expectations about utilization of the game-based learning approach

Statement	1-Strongly Disagree (%)	2-Disagree (%)	3-Not Sure (%)	4-Agree (%)	5-Strongly Agree (%)	Mean	Standard deviation
[T3.1] I would like to have more opportunities to learn using the game-based learning approach	0	4.3	4.3	60.9	30.4	4.17	.72
[T3.2] I prefer using games to learn compared to traditional methods in class	0	4.3	26.1	39.1	30.4	3.96	.88
[T3.3] I would like to be taught all Informatics courses using educational games	0	26.1	43.5	17.4	13	3.17	.98
[T3.4] I would like such kind of games to be available online for easy access	0	0	0	69.6	30.4	4.30	.47

spondents ([T3.2]: 69.5%) prefer using games for learning compared to traditional methods in class. On the other hand, less than one third of students ([T3.3]: 30.4%) would like to be taught all Informatics courses using educational games, while 43.5% of them are not yet sure. Probably this is due to the fact that students did not have any prior experience in game-based learning in other courses and it is difficult for them to understand how games could be utilized in their context. Finally, all the respondents ([T3.4]: 69.6% agree and 30.4% strongly agree) would like such games to be available online for easy access. Therefore, it seems that students hold high expectations for a game-based learning of programming, but they seem to be divided about the use of game-based learning for other Informatics courses.

4.4. Students' Preferences in Arcade Games and Proposals for Improvement

Proposals for improving the game

As part of the project assigned as homework and the questionnaire, students were asked to propose their own ideas in order to improve the Snake game and make it more attractive to students. The following proposals were made by students:

- *Student 1*: “Initially, an extension that can be made is changing *level*, specifically after a time of period *obstacles* can be created for making the game more difficult. Also, according to the score the snake could be passing through the walls and continue to the other side for a short period of time. Furthermore, the *score* could be presented during the whole game. Moreover, the score could be saved and the 5 best scores could be presented. Finally, we could add time for each fruit that appears, for example the apple can be visible for 5 seconds and so on”.
- *Student 2*: “An addition could be showing the *score* during the game and not just in the end. A list of *high scores* could exist and at the end the *place* of the player could be presented. During the game a list of *images* could be used to change the background periodically”.
- *Student 3*: “1st the player could enter his *name* and *score* in the end of the game in order to have some *competition*. 2nd there could be some object like fruits that will *speed* up the snake. 3rd there could be the ability for the player to change the *color* of the snake or the background. 4th the player could select at the beginning of the game its *difficulty* (speed, score)”.
- *Students 4*: “My ideas for the snake and the *appearance* of the game that could improve it is for the snake to have a different shape or to *change shape* after a predefined score limit so as to encourage the player to continue playing. Also, there could be fruits that when they are eaten by the snake the game terminates, which means that these fruits are *forbidden*. Furthermore, there could be a method that when the snake is late in eating the fruits the space that it can move can be *restricted*.”
- *Students 5*: “We could change the *appearance* of the snake and also we could place in the window various *obstacles* for making the game more difficult. Finally, we could add a *soundtrack* during the game.”

Students' comments for improving the Snake game can provide general insights for devising games that engage and motivate students, no matter what the area of the game is. Students consider important features for a game the following: existence of levels; obstacles and forbidden actions for increasing difficulty; calculation of scores, high scores and ranking of each player (competition); changes in the appearance of the game's actor and background; music.

Students' preferences in arcade games

As part of the questionnaire students were asked to rank eight well-known, arcade games from the most interesting to the least interesting on a scale from 1 to 8. The purpose of this classification was to form a global view about students' preferences regarding games in order to understand which ones excite their interest for further study. Seventeen out of the 23 students completed correctly the games' classification. For each game the "votes" received for each possible place (1st, 2nd, 3rd, etc.) were counted. For example, 12 from the 17 students ranked in the first place the packman game. Thus, with the same way we obtained the following results:

- 1st place: Packman
- 2nd place: Break Collision
- 3rd place: Tetris
- 4th place: Snake
- 5th place: Space Invaders
- 6th place: Puzzle
- 7th place: Sokoban
- 8th place: Minesweeper

So, the Snake game that was used in our study is somewhere in the middle of this ranking. Besides that students were motivated and evaluated positively the whole approach, as analyzed in the previous sections.

5. Conclusions

In this paper a study on students' attitudes and perceptions on the potential utilization of arcade games for an enhanced learning of programming was presented. In order to investigate this issue a didactical intervention on an undergraduate programming course was designed. Specifically, the well-known Snake game was selected and educational material consisting of a presentation of the game and a set of exercises for extending the game and making it even more interesting, as well as a questionnaire were prepared. This game was selected for the following reasons: (1) it implements several programming concepts, especially GUI concepts that were of interest for us; and (2) it is a game that has managed to entertain many generations. Our main goal was to transfer our students the message that learning programming can be an entertaining activity. A twenty minute presentation of the game was made in the context of a typical lecture and then students had to play the game, study the relevant material, complete the game-related assignments and answer a questionnaire.

The analysis of students' replies in the questionnaire showed that this teaching approach has the potential to increase students' motivation and interest in programming significantly. Using arcade games for teaching and learning programming can provide a much more creative and entertaining context for students. The majority of students stated that they prefer learning programming concepts through the analysis of games and also they prefer solving game-based programming exercises. Moreover, they believe that if they were taught programming with games they would have performed better in relevant courses.

Utilizing arcade games in problem solving was perceived by the majority of students as both a very interesting and encouraging activity. Students stated that the game strengthened their efforts towards comprehending the underlying concepts and helped them develop critical thinking.

The majority of students prefers games for learning in comparison with typical teaching methods and would like to have more opportunities for learning through games. Students consider games ideal for learning programming and would like to use them for learning programming concepts in the future, but they seem not to be sure if the same could apply to all fields of informatics.

Concluding, it is clear that the results of this study were rather positive regarding the utilization of arcade games for learning programming. However, it is also clear that further research is required in order to validate the results. It is obvious from students' replies that their motivation for learning programming is increased, but their impact on acquiring problem solving skills has to be evaluated. An experimental study with a larger number of participants, more game-based exercises and evaluation of their impact on students' achievements is considered necessary in order to draw more objective and reliable results. Such a study could be based on utilizing games that were ranked in the first places in students' preferences, such as PacMan, Break Collision and Tetris. Moreover, the educational benefits of utilizing networked game engines should be investigated, since students explicitly stated their preference in having such games available online.

References

- Barnes, T., Chaffin, A., Powell, E., Lipford, H. (2008). Game2Learn: improving the motivation of CS1 students. In: *Proceedings of the 3rd International conference on Game development in computer science education*. 1–5.
- Brusilovsky, P., Calabrese, E., Hvorecky, J., Kouchnirenko, A., Miller P. (1997). Mini-languages: a way to learn programming principles. *International Journal of Education and Information Technologies*, 2, 65–83.
- Chen-Chung Liu, Yuan-Bang Cheng, Chia-Wen Huang (2011). The effect of simulation games on the learning of computational problem solving. *Computers & Education*, 57(3), 1907–1918.
- Ibrahim, R., Yusoff, R.C.M., Omar, H.M., Jaafar, A. (2011). Students perceptions of using educational games to learn introductory programming. *Computer and Information Science*, 4(1), 205–216.
- Kazimoglu, C., Kiernan, M., Bacon, L., Mackinnon, L. (2012). A serious game for developing computational thinking and learning introductory computer programming. *Procedia – Social and Behavioral Sciences*, 47, 1991–1999.
- Ladd, B. C., Jenkins, J. (2011). *Introductory Programming with Simple Games: Using Java and the Freely Available Networked Game Engine*. John Wiley & Sons.

- Malliarakis, C., Satratzemi, M., Xinogalos, S. (2012). Towards the constructive incorporation of serious games within object oriented programming. In: *Proceedings of the 6th European Conference on Games Based Learning (ECGBL 2012)*. 301–308.
- Malliarakis, C., Satratzemi, M., Xinogalos, S. (2013). Towards a new massive multiplayer online role playing game for introductory programming. In: *Proceedings of the 6th Balkan Conference in Informatics (BCI '13)*. ACM Press, 213–220.
- Muratet, M., Torguet, P., Viallet, F., Jessel, J.P. (2010). Experimental feedback on Prog&Play, a serious game for programming practice. *Eurographics*, 1–8.
- Muratet, M., Torguet, P., Viallet, F., Jessel, J.P. (2011). Experimental feedback on Prog&Play: a serious game for programming practice. In: *Computer Graphics Forum*. Blackwell Publishing Ltd, 30(1), 61–73.
- O'Kelly, J., Gibson, P. (2006). RoboCode & problem-based learning: a non-prescriptive approach to teaching programming. *ACM SIGCSE Bulletin*, 38(3), 217–221.
- Phelps, A, Bierre, K., Parks, D. (2003). MUPPETS: multi-user programming pedagogy for enhancing traditional study. In: *Proceeding of the 4th Conference on Information Technology Education, October 2003, Lafayette, Indiana, USA*. 100–105.
- Xinogalos, S. (2012). Programming techniques and environments in a technology management department. In: *Proceedings of the 5th Balkan Conference in Informatics (BCI 2012)*. ACM Press, 136–141.

A. Theodoraki holds a Bsc on Technology Management, University of Macedonia, Greece. Her thesis was on Educational Games for Introductory Programming and now works as a software analyst and programmer.

S. Xinogalos is a lecturer of Programming Environments and Techniques at the Department of Applied Informatics, University of Macedonia, Thessaloniki, Greece (e-mail: stelios@uom.edu.gr). He is author or co-author of more than 50 research papers on programming environments and techniques, teaching and learning OOP, didactics of programming and educational technology. He has also designed and implemented two educational programming environments.

Studentų nuostatų taikyti žaidimų programų pirminio teksto pavyzdžius programavimui mokytis tyrimas

Aristea THEODORAKI, Stelios XINOGALOS

Mokymuisi skirti žaidimai dabar dažnai taikomi įvairiuose dalykuose mokinių motyvacijai kelti ir mokymosi patirčiai išplėsti. Šis naujas technologijomis grįstas mokymasis sudomino mokslininkus ir mokytojus, tiriančius ir dėstančius programavimą, kadangi pastarasis yra itin reikšmingas kompiuterių moksle. Iki šiol yra sukurta keletas edukacinių žaidimų ir pirmieji tyrimai atskleidžia jų teigiamą įtaką mokymusi. Straipsnyje pristatomi arkadiniai žaidimai, skirti mokytis programavimo. Remiantis šia metodika mokiniai pirmiausia žaidžia paprastą žaidimą, pavyzdžiui, „Gyvate“ ar „Tetrij“, tada išnagrinėja žaidimo programos pirminį tekstą ir galiausiai praplečia jį. Žvalgomojo tyrimo metu studentai išnagrinėjo gerai žinomo žaidimo „Gyvate“ programą ir praplėtė ją naujais funkcionalumais. Studentų užpildytų apklausų analizė parodė, kad arkadinių žaidimų taikymas programavimo conceptams mokytis didina studentų motyvaciją, padeda jiems suprasti kompleksinius conceptus ir įtraukia atlikti programavimo veiklas.

