Investigating the Perceived Player Experience and Short-term Learning of the Text-based Java Programming Serious Game "Rise of the Java Emperor"

Dimitrios TSIOTRAS, Stelios XINOGALOS

Department of Applied Informatics, School of Information Sciences, University of Macedonia
156 Egnatia Street, GR-54636, Thessaloniki, Greece
e-mail: mai19078@uom.edu.gr, stelios@uom.edu.gr

Abstract. There has been an active movement towards fun learning, using games in education. This article introduces the text-based serious game "Rise of the Java Emperor" that aims to support students in learning basic object-oriented programming concepts using Java. Information concerning the analysis, the design and the pilot evaluation of the game is presented. Thirty-three undergraduate and postgraduate students of an Applied Informatics Department voluntarily played and answered a questionnaire based on the MEEGA+ model, in order to investigate the perceived player experience and short-term learning as well as the acceptance of a text-based programming game by students. The results of the evaluation show that text based games can be both fun and instructional for the field of computer programming. An important issue that requires further research is how this or other programming games can be successfully combined with traditional teaching methods for enhancing the learning of programming.

Keywords: Serious games, educational games, 2D games, text-based games, learning programming, Java.

1. Introduction

Nearly half a century after their appearance, videogames have become an integral part of the daily lives of millions of people around the world. They have also succeeded in being established as a basic choice to anyone who has access to them and knows how to use them for spending their free time. Videogames attract intense scientific interest and research into the games' impact on development and learning, thereby leading to the development of educational games for serious purposes.

Serious games are fun videogames designed for educational purposes that enhance the educational experience, transfer information and knowledge, while developing and

practicing certain skills. They also provide feedback either from the game play itself or from interacting with other players. They can quite often bypass the monotony and boredom of classical textbook study by psychologically transforming the experience of study from feeling tired to an opportunity for fun and relaxation (Zhang and Lu, 2014).

Fields of Computer Science and programming in specific could not be left behind, as researches on whether or not serious games can introduce the player to programming have already been carried out (Leutenegger and Edgington, 2007). More and more games are being developed with the ultimate purpose of teaching the players the basic principles of a programming language. What needs to be investigated in depth is whether or not these games are successful in teaching, gaining the interest of the players and attracting them to learn a new programming language (Muratet *et al.*, 2009). The serious game Rise of the Java Emperor, which is presented in this article, was developed as an attempt to teach players the basic concepts and elements of the Java programming language. Java is an object-oriented programming language that has attracted the interest of instructors and students for several years. The purpose of this educational game is to give players the necessary motivation to get in touch with the basic elements of the Java programming language, such as *variables*, *objects* and *classes* and, as the game proceeds, with more complicated concepts, such as *methods*, *arrays*, *conditional statements* and *loops*. The fact that this game starts with the very basic concepts of programming gives the opportunity to every user to play it, since there is no need for prior programming knowledge. However, it is clear that the game alone is not an adequate means of learning programming, but rather an additional means of getting prepared or as a supplementary means of learning in the context of a traditional learning setting. The main research question of this study is the following:

What is the perceived player experience and short term learning in the text-based programming game Rise of the Java Emperor?

We anticipate that the results of this research question will also give us some preliminary hints on the effectiveness of text-based serious games in the field of programming.

The rest of the article is organized as follows. In section 2, we briefly review existing serious games for object-oriented programming and Java, as well as the rationale behind the implementation of the serious game Rise of the Java Emperor. In section 3 the analysis and design of the game is presented using the *Conceptual Framework for Serious Games* proposed by Yusoff, Crowder, Gilbert & Wills (2009). Section 4 briefly presents key elements regarding the implementation of the game. A detailed description of the pilot evaluation carried out regarding its methodology, the participants and the results is presented in section 5. Finally, we conclude and suggest plans for future research in section 6.

2. Background

Over the years, a variety of serious games for supporting the teaching of programming languages has been developed. Some of these games aim to teach programming concepts to primary school students, (Giannakoulas and Xinogalos, 2018), while others aim

to support undergraduate students when studying programming at University (Mathew *et al.*, 2019; Malliarakis *et al.*, 2017).

In Table 1, we present and review several non commercial serious games that have been developed through the ages with teaching the basics of Java as their main goal. These games have been developed by the Educational Technology Research Group at the University of Macedonia[1] with the support of undergraduate and postgraduate stu-

Table 1

Serious Games for Programming Using Java

Serious game	Description
jAVANT-GARDE (Galgouranas and Xinogalos, 2018)	A *2D platform* game that aims to support the teaching of basic programming concepts of Java to students. The player controls a robot to explore the level and perform various tasks to complete it in order to move on to the next one. The programming framework that was utilized for its development is the Java LibGDX Framework. The IDE that was used for the development of this game was Android Studio (`https://developer.android.com/studio/`). The game is open source and can be utilized and extended by anyone interested. Available at: `https://sites.google.com/a/uom.edu.gr/stelios-xinogalos/serious-games/programming`
JavabotWars	A *3D first-person shooter serious game* for supporting the player in acquiring object-oriented programming knowledge in Java. The hero of the game, controlled by the player, is a human who, due to an accident in war, has been added robotic parts. Its robotic parts work with Java. The player must stop the war that has been broken out between robots and humans. The game was developed with the game engine Unreal Engine 4. Available at: `https://sites.google.com/a/uom.edu.gr/stelios-xinogalos/serious-games/programming`
The game of Code: Lost in JavaLand	A 2D game for learning object-oriented programming concepts using Java. The player controls a character named Alex who aims to save the destroyed planets and defeat the evil character of the game, named DragonDrop. To achieve this, the player has to use Java commands in order to fight and save the planets. The game was implemented in the development environment Greenfoot (`https://www.greenfoot.org/door`), with the help of the software development platform GitHub. This game aims primarily at primary school students and does not require any programming knowledge. The game is open source and can be utilized and extended by anyone interested. Available at: `https://sites.google.com/a/uom.edu.gr/stelios-xinogalos/serious-games/programming`
Extension of the microworld objectKarel	A 3D programming game for learning object-oriented programming in Java. The game was based on the programming microworld Karel the Robot (Pattis *et al.*, 1995) and its implementation of objectKarel (Xinogalos *et al.*, 2016). The player programs a robot in the Java. The main goal for the player is to avoid some objects and to reach the terminal gate in order to pass to the next level. The game engine that was used for its development is JMonkey 3.0 (`https://jmonkeyengine.org/`). Available at: `https://sites.google.com/a/uom.edu.gr/stelios-xinogalos/serious-games/programming`

---

[1] `https://sites.google.com/uom.edu.gr/edtechrg/homepage`

dents. The ultimate goal of these games is twofold. Firstly, we want to provide students with different genres of Java games, in order to satisfy the varying preferences of students in specific game genres. Secondly, we want to investigate what game genres are appropriate for serious games targeted to the field of computer programming. So far, a 3D game based on the *microworld* Karel the Robot (Pattis *et al.*, 1995) and the features of objectKarel (Xinogalos *et al.*, 2006), a *platform game* called jAVANT-GARDE (Galgouranas and Xinogalos, 2018), a *3D first-person shooter game* called JavaBot Wars and a set of *2D mini games* called "The game of code: Lost in JavaLand" have been developed. From these games, the only one that has been evaluated is jAVANT-GARDE. jAVANT-GARDE was positively evaluated by secondary education students in terms of perceived player experience and short term learning using the MEEGA+ model (Petri, von Wangenheim & Borgatto, 2016).

The proposed game **Rise of the Java Emperor** aims to provide *a text-based game* that can be used by anyone who is interested in learning the basics of Java programming in a pleasant and interesting way. Text-based games feature rich storytelling and require language understanding, skills like long-term memory and planning, exploration and common sense (Côté, 2018). Based on the aforementioned features, text-based games were considered appropriate for the field of programming. Our search for text-based programming games brought to our notice the educational game SQL Island (Schildgen, 2014). SQL Island is a text-based adventure programming game for learning the structured query language SQL that is used for communicating with relational database management systems. Although we were not able to find student evaluation results for this game, it must be noted that SQL Island was awarded in the web-based category of the 2nd International Educational Games Competition carried out at the European Conference on Games-Based Learning in 2014. These facts provided the motivation for designing **Rise of the Java Emperor** and carrying out the pilot evaluation presented in this article.

3. Analysis and Design of Rise of the Java Emperor

The serious game Rise of the Java Emperor aims to educate the player on the basic concepts of programming with Java, by using a story as a means of transmission to make this happen. The theory of Java is organized into *eight different levels* in the game and each one of them covers different programming concepts, in order for the learning goals to be distributed in a clear and structured manner. The player controls a character named Trip, explores each level and meets different non player characters, who help the player learn the basics of programming in Java and test the player's knowledge by giving different quizzes to complete. After that, in some occasions, the player must face the opponent character and defeat him in order to continue to the next level. In this section, basic information about the game is presented (section 3.1) and design decisions are analyzed in the context of the *Conceptual Framework for Serious games* proposed by Yusoff, Crowder, Gilbert and Wills (2009) (section 3.2).

3.1. *Basic Data*

Instructional Objectives

- Raise students' interest in programming by presenting and explaining basic concepts of programming, including variables, objects, classes, methods, arrays, loops and conditions.
- Familiarize students with basic concepts of programming by making them write code and answer several questions about programming concepts.
- Support students in self-learning programming and self-assessing their knowledge.

Game Objectives

The game Rise of the Java Emperor is a 2D game that combines programming and battles. In this game, the player controls a young explorer named Trip and guides him through an interesting trip around the Java Island. Trip's quest is to defeat every Java Captain and learn as much as he can about Java, so he can defeat the Java Emperor and become his successor. Every level introduces a new Java concept and each question has to do with the game's story.

The main structure of each level (Fig. 1) is the following:

- The player meets a non-player character, who is the master of a specific concept in Java (Fig. 2).
- After studying theory on the help panel and reading what the Java Captain has to say, the player must complete several programming quizzes that the Captain gives, in order to test the player's knowledge (Fig. 3).
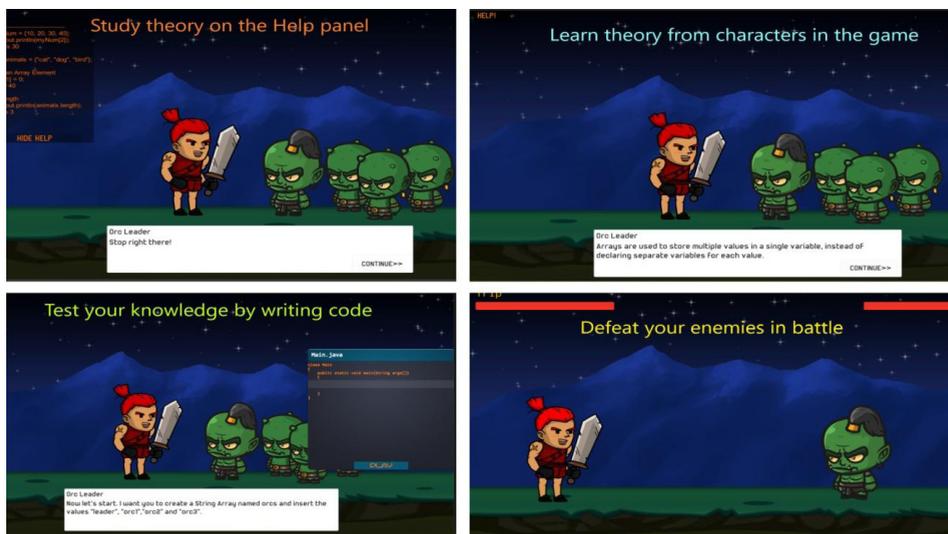

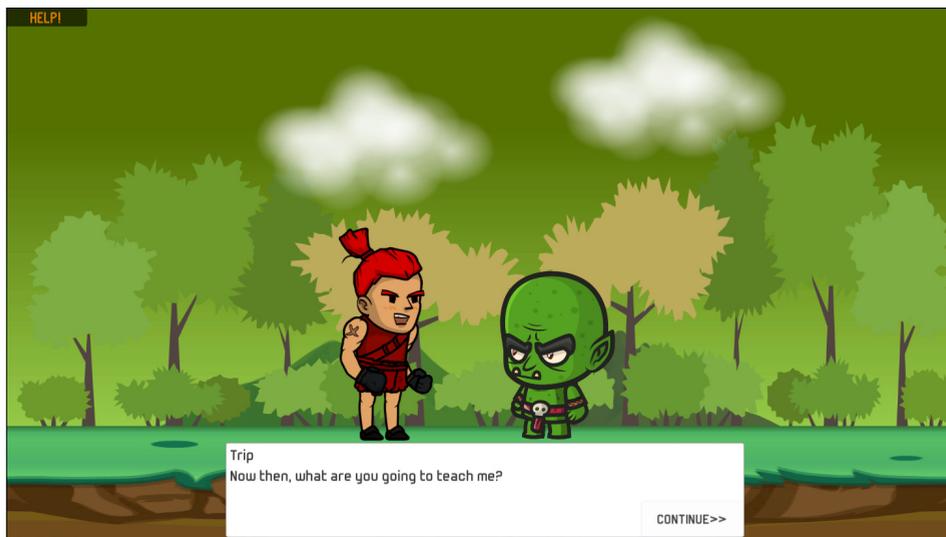Fig. 1. Main structure of Rise of the Java Emperor.

Fig. 2. The player meets the Java Captain.

Fig. 3. The player studies theory.

- To complete the quests, the player is asked to complete Java classes by writing code inside blank spaces (Fig. 4).
- When the player completes all the level's quests, s/he moves on to the next level. In some levels, the player must battle the opponent character and defeat him in order to continue to the next level (Fig. 5).
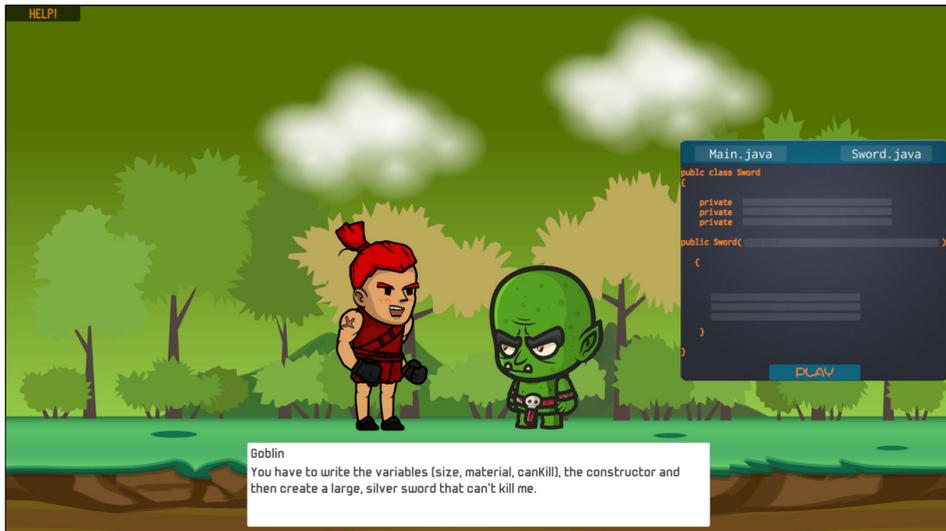
Fig. 4. The player answers the enemy's quizzes.

Fig. 5. The player battles with an enemy.

Target Audience

The game does not demand any previous knowledge on programming and anyone interested in it may benefit from the game. However, it is expected to be more effective if used as a supplementary means of comprehending object-oriented programming concepts and Java syntax in the context of a traditional learning setting.

Preparation and Set-Up Time

Currently, there is only a desktop version of this game for Windows, which has to be downloaded to the players' computers in order to be played. For a good player experience a resolution of 1360 × 768 or higher must be selected through the game's Options menu.

The URL for downloading "Rise of the Java Emperor" is:

```
https://sites.google.com/a/uom.edu.gr/stelios-xinogalos/serious-
games/programming#TOC-Rise-of-the-Java-Emperor:-a-2D-serious-game-
for-learning-the-basics-of-Java
```

Playing Time

The playing time varies from player to player, depending heavily on the prior knowledge of programming, as well as the difficulty that s/he faces in understanding the concepts that are introduced. Furthermore, the playing time depends on the players' experience with this kind of games and how easily they adjust to the rules of the game. The average players' playing time, based on the results of the pilot evaluation of the game, is between 1 to 2 hours in order to complete all the levels and battles of the game, but it depends on the aforementioned factors.

3.2. *Game Design, Pedagogy & Learning Content Modeling*

As already mentioned, Rise of the Java Emperor was designed based on the *Conceptual Framework for Serious Games* proposed by Yusoff, Crowder, Gilbert & Wills (2009). This framework includes learning pedagogy theory in combination with gaming requirements and targets to establish a conceptual model that is used by game designers or educational practitioners when they design serious games for effective learning. In Table 2 we present the major components that create an effective model for learning through the use of serious games, according to the authors of *Conceptual Framework for Serious Games* (2009), along with the way we approached them in our game.

Table 2

Components of Yusoff's *et al.* (2009) Conceptual Framework for Rise of the Java Emperor

Components	Design decision
Capability	The game is structured in a way that the player can remember every basic programming concept in Java based on the level that s/he played. For example, when the player wants to remember how to create an object in Java, s/he will recall the level that s/he had to create a ship or the level that s/he created a sword. Through the game, the player will develop skills in object-oriented programming, logic and attention to detail.

Table 2 – continued from previous page

Components	Design decision
Instructional content	At each level the player learns about different concepts and Java principles. Each Java Captain presents in detail a principle in Java, gives specific examples and tests the player by giving quizzes to complete. To put it simply, every Java Captain is a teacher that specializes in a different element in Java, like variables (level 1), objects (levels 2–3), methods (level 4), arrays (level 5), loops (level 6) and conditions (level 7).
Intended learning outcomes	In this game, the player is constantly tested on what s/he has learned by writing code to answer the questions. By coding, the player pays more attention to learn about the basics of Java programming language and tests the acquired knowledge in practice, by writing the right code parts inside empty fields, so s/he can complete every class correctly and continue on to the next level. Thus, the intended learning outcomes of this game include the comprehension of the basic principles of Java theory, the familiarization with the subject and the achievement of combinatorial problem solving skills.
Game attributes	The game is designed in such a way that every player can learn at his/her own pace. First of all, the player can skip levels and choose the level and the principle in Java that s/he wants to study. Furthermore, short-term memory of the player has been taken into consideration and every level is designed in such a way that the learner does not have to remember everything. Moreover, help is given to the player in every quiz that s/he has to answer and after completion, the player is rewarded with a different reward each time. Finally, more difficult principles in Java are presented more than once to the player (for example: objects), thus making him/her practice more and understand in depth the element that s/he studies, since the difficulty of each task is harder every time.
Learning activity	Each Java Captain that the player encounters teaches him/her about a basic principle in Java. Besides teaching the player about basic principles, the Captain mentions several examples, so that the player can understand completely the proper use of the principle that s/he learns. Furthermore, if the player gives a wrong answer, the game shows a message that informs the player that a specific answer that s/he gave is wrong. Moreover, every level that the player plays is slightly more difficult than the previous one. Finally, many levels demand from the player to use the knowledge that was gained in a previous level, thus making the player try harder in order to complete every level.
Reflection	The strategy that the player applies is determined by the game itself. The player has to learn about each Java element and complete specific quizzes by writing the right parts of code. However, there may be times where the player gives a wrong answer. The game provides many explanations of each element so that the player can learn more about each concept and not repeat the same mistakes.
Games genre	This game does not belong to a specific game genre, but it belongs to a combination of game genres. It combines: programming while the player completes *quizzes*; *text based adventure* through dialogues between the enemies and the main character; and *battles* of the player against several enemies.
Game mechanics	Since the game combines programming and battles, the game mechanics differ for each state of the game. While the player completes programming quizzes, s/he has to read what the Java Captain has to say, read the Help panel if s/he wants to and complete each class with the right code parts. When it comes to a battle, the player has to find the weak point of the enemies and attack until the opponent's life points reach zero, while taking into consideration the main character's life points.
Game achievement	The achievements in this game are the total amount of resources and assets that the player collects. When the player completes a quiz correctly, the game awards him/her with things such as a ship that s/he can use to pass through the lake, a sword that the player can use to fight and many more.

4. Implementation of Rise of the Java Emperor

Rise of the Java Emperor was implemented using the programming language *C# (C Sharp)*. Like Java, C# is an object-oriented programming language based on the *.NET Framework*.

   One of the most crucial steps in the implementation of the game in the early stages was to determine which tools were to be used for the development of the game. Examining the features and reviews of various game engines and game development frameworks, led to the decision that the Unity engine was appropriate for the implementation of the game Rise of the Java Emperor. Unity (`https://unity.com/`) is a cross-platform game engine developed by Unity Technologies, first announced and released in June 2005. The Unity engine has many advantages that anyone can use in order to develop different games. To start with, the basic Unity model is free. Furthermore, it is very easy for the creator to program on this platform and use his/her code as s/he wishes, since this engine provides unlimited possibilities for creation. Finally, the Unity Engine's *Asset Store* provides many free environments, sound effects and characters which can be used from everyone who wants to develop a game.

   Another stage of implementation, which proved to be time-consuming, was the search for suitable and attractive assets. Therefore, the audio and the graphics of the game were carefully selected among a large pool of free assets that are available on the web and on the Unity's Asset Store. A lot of effort was put into combining the assets together, while a lot of editing was required as well. The majority of graphics was taken from two specific websites: `https://assetstore.unity.com`, `https://craft-pix.net/freebies/`. As far as the game's music is concerned, it was taken from the website `https://freesound.org/`.

5. Pilot Evaluation of Rise of the Java Emperor

The pilot evaluation of Rise of the Java Emperor was carried out through an anonymous questionnaire that was completed by the participants, after their experience with the game. In this section, we present the methodology, the demographic data of the participants and the results of the pilot evaluation.

5.1. Methodology

The participants of the study voluntarily played Rise of the Java Emperor and filled in a questionnaire based on the MEEGA+ model for evaluating the game. The main purpose of the MEEGA+ model is to analyze educational games, in order to evaluate the quality of player's experience and whether the learning objectives are achieved or not (Petri *et al.*, 2016).

According to the model, the player experience axis is split into the following dimensions: focused attention, relevance, confidence, satisfaction, usability, social interaction, challenge and fun. On the other hand, the axis of perceived learning is split into short-term learning and learning goals.

The final questionnaire that was designed based on the MEEGA+ model consisted of three sections containing 49 questions in total:

- Demographics (13 questions).
- Player experience (29 questions).
- Perceived learning outcomes (7 questions).

The questionnaire did not include questions regarding the social interaction dimension of the MEEGA+ model, since the game does not support such interaction. The questions were to be answered on a Likert-scale from −2 to 2 (−2 = strongly disagree, −1 = disagree, 0 = indifferent, 1 = agree, 2 = strongly agree). Utilizing the proposed MEEGA+ model, the median and the percentages for each possible response in each question were calculated, so that some conclusions can be drawn.

5.2. *Participants*

The game was evaluated by undergraduate and postgraduate students of an Applied Informatics Department. Thirty-three participants responded to an invitation (sent through email) to play and evaluate Rise of the Java Emperor in a time period of fifteen days. The participants anonymously filled in the questionnaire that was prepared using Google forms. This research has been conducted following the ethical requirements established by Greece national board of ethics.

5.3. *Results and Discussion*

Demographic information. Fig. 6 presents some demographic information for the 33 participants of the pilot evaluation. The answers show that the participants have varying programming knowledge, with 27% of them having little or no knowledge of procedural programming and 46% of them having little or no knowledge of object oriented programming. Moreover, 70% of the participants managed to beat the game, which indicates that the game might be difficult for the players or not fun enough. Finally, the results show that the average playing time is between 1 to 2 hours, since 40% of the participants played the game for 1 hour and 33% of them for 2 hours. Taking into account the 23 out of the 33 participants that completed the game, 39% of them completed the game in one hour and 39% of them in 2 hours, which means that the average time for completing the game is approximately 1.5 hours.

Usability. Fig. 7 presents the results of the evaluation regarding the usability of Rise of the Java Emperor. Nearly all the participants (97%) agree or completely agree that the game design (interface, graphics, etc.) is attractive, while the fonts and colors of the
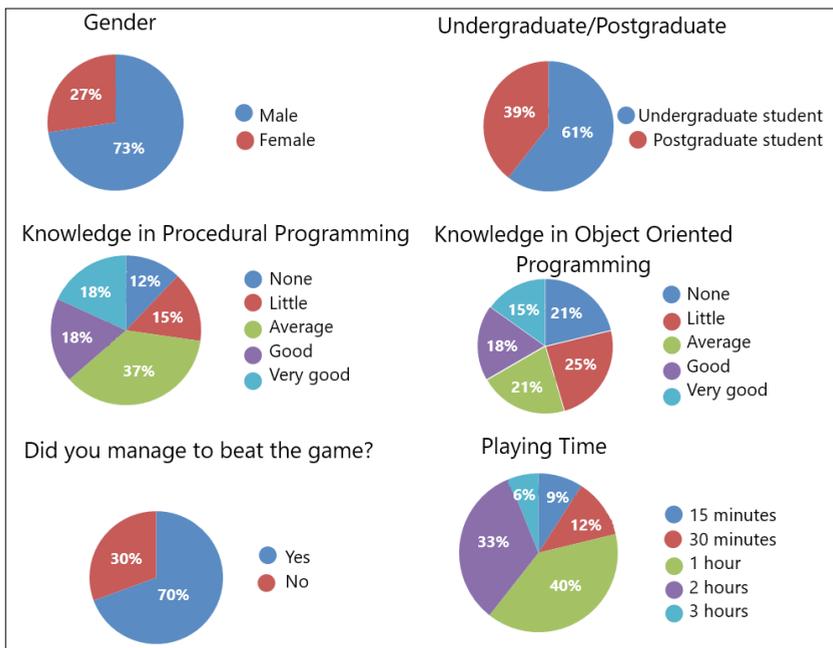
Fig. 6. Demographic data of the pilot evaluation.
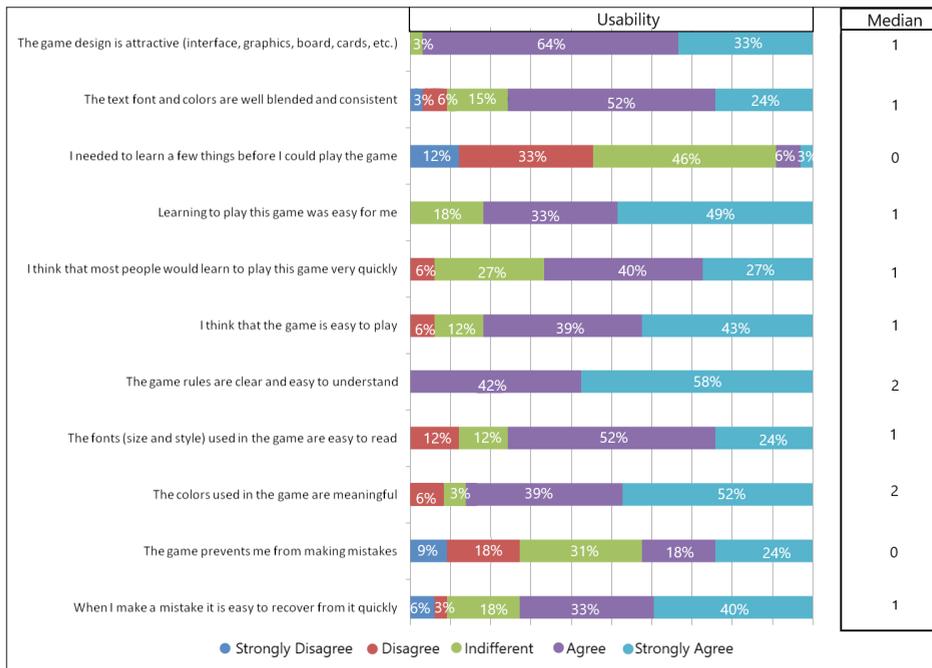

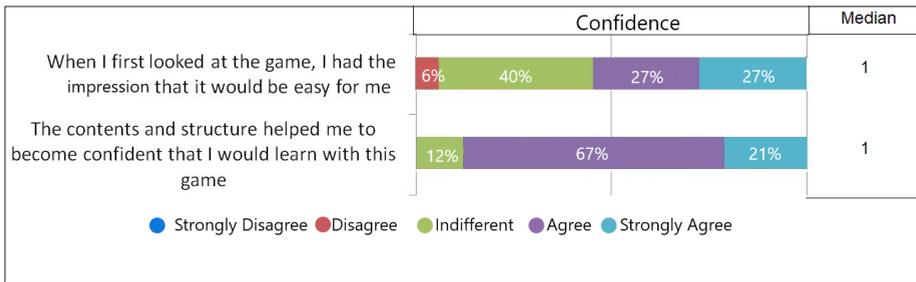Fig. 7. Evaluation results on Usability.
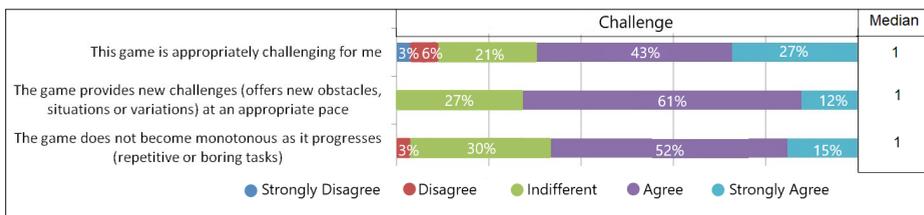
Fig. 8. Evaluation results on Confidence.

Fig. 9. Evaluation results on Challenge.

game were also rated positively. Furthermore, 82% of the participants agree or completely agree that they easily learned to play the game, while 67% of them think that most of the players would learn to play this game very quickly, 82% of them believe that this game is easy to play and all of them think that the game rules are clear and easy to understand. Moreover, 42% of the participants agree or completely agree that the game prevents the players from making mistakes and 73% of them think that when the player makes a mistake it is easy to recover from it quickly.

Confidence. Fig. 8 presents the results of the evaluation regarding the confidence of the participants. The majority of the participants (54%) agrees or completely agrees that the game would be easy by the first look, while 88% of them agree or completely agree that the contents and structure helped players to become confident that they would learn with this game.

Challenge. The results of the evaluation regarding the challenging aspects of the game are presented in Fig. 9. Nearly three out of four participants found the game challenging for them (70%) and believe that the game provides new challenges at an appropriate pace (73%) while 67% of them believe or strongly believe that the game does not become monotonous as it progresses.

Satisfaction. Fig. 10 presents the results of the evaluation regarding the players' satisfaction. The results exceeded our expectations, with more than three out of four participants feeling satisfied after completing the game tasks (median 1 – agree), while 61% of the participants agree or completely agree that they advanced in the game due to
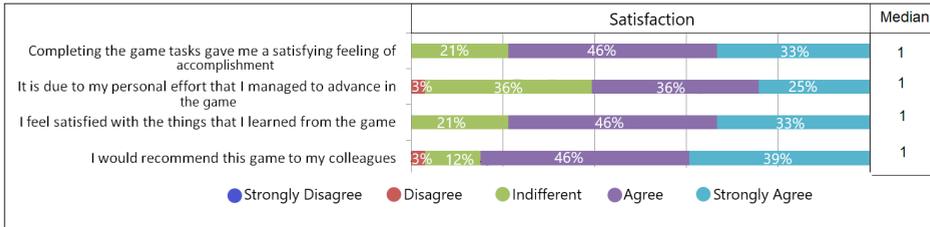
Fig. 10. Evaluation results on Satisfaction.

their personal effort. Furthermore, 79% of the participants feel satisfied with the things that they learned from the game (33% completely agree – 46% agree), while the most positive response was that 85% of the participants would recommend Rise of the Java Emperor to their colleagues.

Fun. The results regarding the fun elements of the game are presented in Fig. 11. Specifically, 79% of the participants (55% agree – 24 % completely agree) had fun playing the game, while 73% of them smiled because of something that happened during the game.

Focused Attention. Fig. 12 summarizes the results of the evaluation regarding the focused attention of the players while playing the game. Half of the participants found something interesting at the beginning of the game that captured their attention. However, more than half of the participants disagree that they forgot their surroundings while playing the game, while 36% of them were not so involved in the gaming tasks that they lost track of time. These results show that the game can be improved in order to achieve and keep more and more users focused.
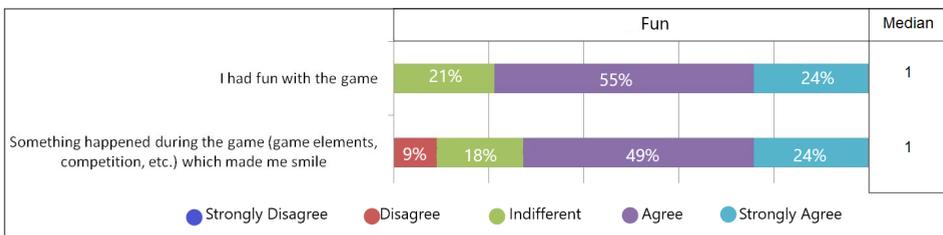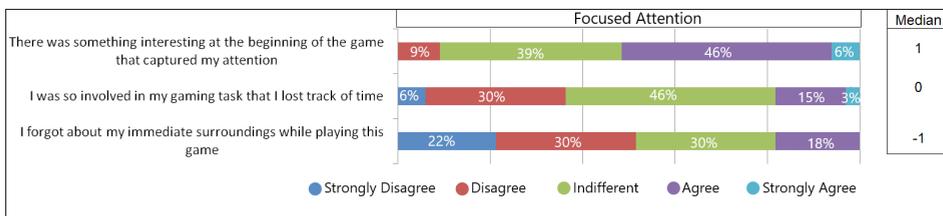

Fig. 11. Evaluation results on Fun.

Fig. 12. Evaluation results on Focused Attention.

Relevance. The results of the evaluation regarding the relevance of the game with programming are presented in Fig. 13. The answers that were given were particularly positive, with 97% of the participants believing that the contents of this game are related to object-oriented programming (median 2 – strongly agree), while 90% of them think that this game is an adequate teaching method for object-oriented programming (64% agree – 27% strongly agree). Moreover, more than 50% declared that the game contents are relevant to their interests (median 1 – agree). Finally, the majority of the participants prefer learning with this game to learning through other ways (median 1 – agree).
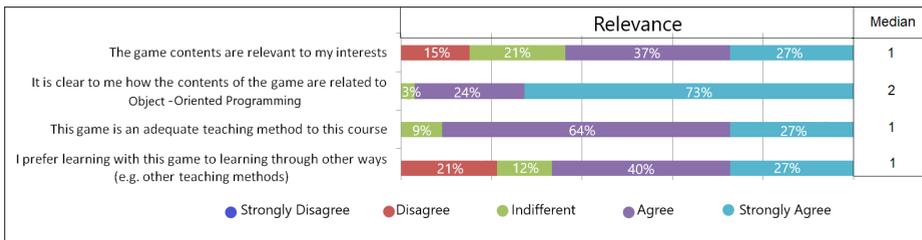
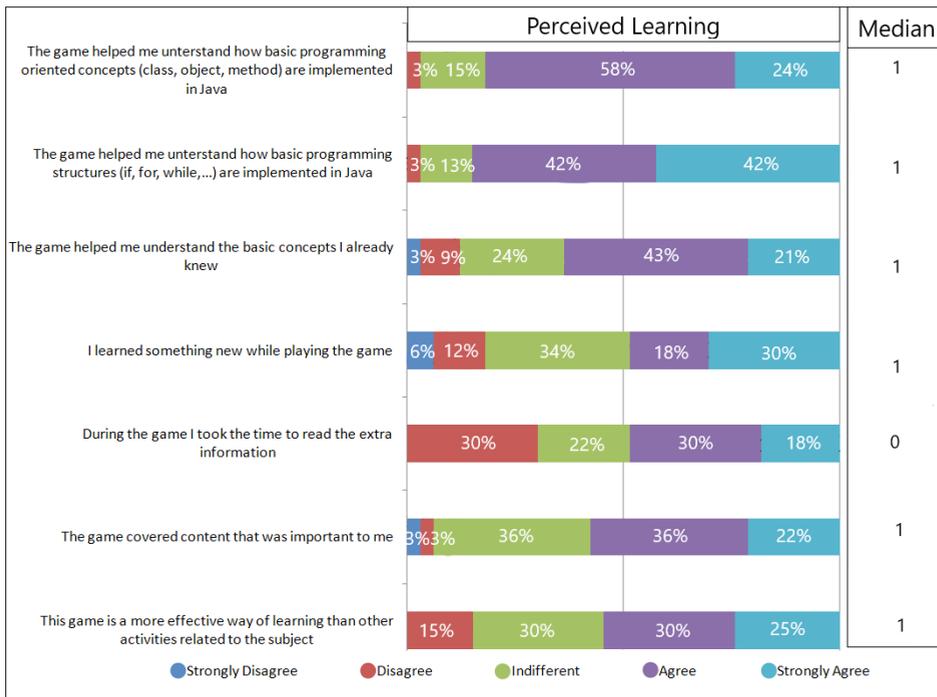

Fig. 13. Evaluation results on Relevance.

Fig. 14. Evaluation results on Perceived Learning.

Perceived learning. Fig. 14 contains the results of the evaluation that refer to the perceived learning of the players after their experience with the game. Eight out of ten participants agree or strongly agree that this game helped them understand the basic concepts of Java, such as objects, variables, conditions and loops. Moreover, over half of them agree or strongly agree that this game helped them to understand the basic concepts they already knew. Finally, over 50% of the participants believe that this game is allowed for efficient learning compared with other teaching/learning activities (median 1 – agree) and, also, that the game covered content that was important to them (median 1 – agree). This shows that the game has succeeded in achieving its goal of teaching Java.

6. Conclusion

Rise of the Java Emperor is a new 2D text-based adventure serious game, developed for introducing beginners to basic concepts of the Java programming language. In terms of player experience, the results of the pilot evaluation were largely satisfying. The category with the less positive results which shows that there is room for improvement is the one about focused attention. On the other hand, the results in terms of short-term learning were all positive, with some room for improvement as well. Specifically, the majority of the participants consider that this game is ideal for learning the basic concepts of the programming language Java. However, they also gave mixed answers as to whether such ways of learning programming are more efficient than the traditional ones. Learning programming requires a lot of practice on writing code and it is clear that a game cannot substitute traditional teaching methods, but it can enhance them if it is carefully designed taking into account various pedagogical issues. The fact that this game was evaluated by players with both a different level of prior programming knowledge in Java and varying ages and, still, the majority evaluated the game positively shows that videogames can ultimately motivate students in learning programming. However, this game has to be played and evaluated by more students and its effects on learning programming have to be thoroughly studied, in order to extract safer conclusions regarding its effectiveness as a learning tool.

Rise of the Java Emperor has many similarities in terms of gameplay with the game SQL Island (Schildgen, 2014). The fact that this game is also text-based shows that, even nowadays, text-based games can still be fun and educational at the same time. Both games present theory and multiple examples in order for the player to learn more easily programming. Furthermore, both games require coding by filling in gaps on their own from the very beginning. Such facilities provide the opportunity to code, experiment and learn in a friendly programming environment, while the player enjoys the game's storytelling. Although further research is required we have strong implications that text-based games are a good choice for the field of programming.

In conclusion, the results of the evaluation show that this game was perceived as useful by students, and with the right adjustments it can become even more effective and achieve even bigger goals. For example, the game could support players with different programming experience levels (beginner, introductory, median, expert) and could even incorporate a compiler for giving the chance to implement complete programs. Cross-platform compatibility is also important, and a version for Linux which is a widely used platform is considered necessary. An interesting issue that requires further research, besides the investigation of the game's actual impact on learning programming, is how it could be used in classroom in combination with traditional teaching methods for an effective teaching and learning of programming.

References

Côté, M,, Kádár, A., Yuan, X., Kybartas, B., Barnes, T., Fine, E., Moore, J., Hausknecht, M., Tao, R., El Asri. L., Adada, M., Tay, W., Trischler, A. (2018). TextWorld: A Learning Environment for Text-based Games. *Computer Games Workshop at IJCAI 2018*, Stockholm. `arXiv:1806.11532v2`

Galgouranas, S., Xinogalos, S. (2018). jAVANT-GARDE: A Cross-Platform Serious Game for an Introduction to Programming with Java. *Simulation & Gaming*, 49(6), 751–767. DOI: 10.1177/1046878118789976.

Giannakoulas, A., Xinogalos, S. (2019) Current Trends in On-line Games for Teaching Programming Concepts to Primary School Students. In: Tsitouridou M., A. Diniz J., Mikropoulos T. (Eds) Technology and Innovation in Learning, Teaching and Education. TECH-EDU 2018. *Communications in Computer and Information Science*, vol 993. Springer, Cham.

Leutenegger, S., Edgington, J. (2007). A Games First Approach to Teaching Introductory Programming. 38th SIGCSE Technical Symposium, *Computer Science Education*, 115–118.

Malliarakis, C., Satratzemi, M., Xinogalos, S. (2017). CMX: the effects of an educational MMORPG on learning and teaching computer programming. *IEEE Transactions on Learning Technologies*, 10(2), 219–235.

Muratet, M., Torguet, P., Jessel, J. P., Viallet, F. (2009). Towards a serious game to help students learn computer programming. *International Journal of Computer Games Technology*, 3.

Pattis, R. E., Roberts, J., Stehlik, M. (1995). *Karel – The Robot, A Gentle Introduction to the Art of Programming*, 2nd edition. Wiley, New York.

Petri, G., von Wangenheim, C., Borgatto, A. (2016). *MEEGA+: An Evolution of a Model for the Evaluation of Educational Games*. Brazilian Institute for Digital Convergence. Technical Report INCoD/GQS.03.2016.E

Schildgen, J. (2014). SQL Island: An Adventure Game to Learn the Database Language SQL. In: *Proc. The 8th European Conference on Games Based Learning (ECGBL 2014)*, Berlin, Germany, pp. 137–138.

Xinogalos, S., Satratzemi, M., Dagdilelis, V. (2006). An Introduction to object-oriented programming with a didactic microworld: objectKarel. *Computers & Education*, 47(2), 148–171.

Yusoff, A., Crowder, R., Gilbert, L., Wills, G. (2009). A conceptual framework for serious Games. In *Ninth IEEE 2009 International Conference on Advanced Learning Technologies*. IEEE, pp. 21–23. DOI: 10.1109/ICALT.2009.19

Zhang, J., Lu, J. (2014). Using Mobile Serious Games for Learning Programming. INFOCOMP: The Fourth International Conference on Advanced Communications and Computation (pp. 24–29), 2014. DOI: 0.13140/2.1.2779.9689

Mathew, R., Malik, S. I., Tawafak, R. M. (2019). Teaching Problem Solving Skills using an Educational Game in a Computer Programming Course. *Informatics in Education*, 18(2), 359.

D. Tsiotras holds a Bachelor and Master degree since 2020 from the Department of Applied Informatics, University of Macedonia, Thessaloniki, Greece. His interests include programming and (serious) games. The serious game Rise of the Java Emperor was designed and implemented in the context of his master thesis.

S. Xinogalos is an associate professor at the Department of Applied Informatics at the University of Macedonia, Thessaloniki, Greece. He is a member of the Software and Data Engineering Lab, as well as the Educational Technology Research Group at the University of Macedonia, Greece. His research interests include Programming Environments and Techniques, Object-Oriented Design and Programming, Didactics of Programming, Educational Environments and Games for Programming, and Serious Games.