

PY-RATE ADVENTURES: a 2D Platform Serious Game for Learning the Basic Concepts of Programming with Python

Abstract

Background. **Learning programming** is a cognitively demanding field of study accompanied with various difficulties. Although there is a high demand in the market for programmers, software analysts and engineers, a high dropout rate is recorded in relevant fields of study. Serious games are a promising means of engaging students in learning programming by giving them more incentives and making the process of learning programming concepts and languages more entertaining.

Aim. This article introduces a new **serious game** called PY-RATE ADVENTURES, which aims to assist young students in their introduction to the basic **programming concepts** using **Python**. The game does not have any prerequisites and is suitable for players with no previous knowledge of programming. This article aims to present important information regarding the analysis, **design** and **pilot evaluation** of PY-RATE ADVENTURES.

Method. The game was evaluated by 31 people that had recently graduated or were students of an Interdepartmental Programme of Postgraduate Studies in Information Systems. The participants voluntarily played the game and answered a questionnaire based on the **MEEGA+ model**, after their hands on experience with the game. This questionnaire's purpose was to evaluate PY-RATE ADVENTURES in terms of **perceived player experience** and **short-term learning**.

Results. The participants positively evaluated the game almost in all the elements of player experience. Furthermore, the majority of the users consider that the game helped them to learn basic programming concepts in Python and stated that they would prefer to learn programming with this game rather than other teaching methods.

Conclusion. The positive results of the pilot evaluation give us the motivation to proceed and evaluate the game with students in secondary education, in order to extract stronger and generalisable conclusions regarding the impact of the game as an educational tool for learning programming concepts.

Keywords

Serious games, educational games, learning programming, perceived player experience

Introduction

Computer programming is considered an important skill in our society, since we are surrounded by various devices that have to be programmed in order to function appropriately. With the advent of Internet of Things and the interconnection of smaller and bigger devices the ability to program and control them is considered even more important. Efforts to promote *computational thinking* (Wing, 2006), even from young ages, are made around the world and it is usually accomplished through learning programming. However, learning programming is a cognitively demanding task that is accompanied with various difficulties. Besides the importance of programming and the high demand in the market for programmers, software analysts and engineers, a high dropout rate is recorded in relevant fields of study.

Serious games are a promising means of engaging students in learning programming by giving them more incentives and making the process of learning programming concepts and languages more entertaining. These games mostly help students or people with no previous experience in programming to learn basic concepts of programming and develop an algorithmic way of thinking, by focusing on their behaviour during the game and giving them appropriate motives such as rewards, duels and rankings in order to gain their interest and participation (Combefis et al., 2016). After some years of research into the effectiveness of serious games in the field of education, it appears that students find more motivation through games to develop their skills and achieve better results (Stege et al., 2011).

The serious game Py-rate Adventures, which is presented in this article, was developed as an attempt to teach users the basic concepts of programming with the programming language of Python. Python is a programming language that has attracted instructors' and students' interest the last few years. The purpose of this game is to

give users motivation to get acquainted with basic concepts of programming such as *variables* and *operators* and as the game proceeds with more complicated concepts, such as *conditional statements*, *loops*, *classes* and *objects*. The fact that the game starts with the very basic concepts of programming gives the opportunity to everyone to play it, since no prior programming knowledge is needed.

The rest of the article is organized as follows. In section 2, we present a brief and concise review of serious games targeted to an introduction to programming concepts using Python. In section 3, we present the analysis and design of the serious game *Py-rate Adventures*, which is based on the *Educational Games Design Model* proposed by Ibrahim and Jaafar (2009). In section 4 some key elements regarding the implementation of the game are analysed, while in section 5 the results of a pilot evaluation based on a survey utilizing the MEEGA+ model (Petri et al., 2016) that was carried out with the aim of investigating the potential of this new serious game are presented. Finally, conclusions and plans for future research are presented.

Background

Several serious games have been developed for supporting the teaching and learning of programming. Some of these games aim to promote computational thinking at primary school students (Giannakoulas & Xinogalos, 2018), while the goal of others is to familiarize secondary education students with programming concepts (Galgouranas & Xinogalos, 2018), or supporting undergraduate students in dealing with the difficulties encountered when studying programming at University (Malliarakis, Satratzemi & Xinogalos, 2017). In Table 1 we briefly review serious games that have relevant goals and use as a programming language Python.

Py-rate Adventures, the game introduced in this article, aims to provide a platform game that can be used by anyone interested in learning the most fundamental programming concepts using Python as a programming language. Platform games are games that attract players of any age, have simple and straightforward rules and are ideal for introducing novices to programming (Galgouranas & Xinogalos, 2018).

Table 1. Serious games for programming using Python.

Serious game	Description
Rapid router	<ul style="list-style-type: none">• A <i>puzzle game</i> that aims to support the teaching of basic programming concepts to young students.• The player drives a delivery van on a road with bends and lanterns along the way, with the aim to reach a destination point and complete the delivery.• The game is based on the Blockly library by Google, such as the best-known programming environment Scratch. The players move the van by dragging and dropping visual block commands.• The last two levels of the game provide an introduction to the programming language of Python.• Available at: https://www.codeforlife.education/rapidrouter/
CodeCombat	<ul style="list-style-type: none">• A 2D HTML5 browser-based game that aims to support students in acquiring coding skills using either Python or JavaScript.• CodeCombat is a <i>role-playing game</i> and the user controls a hero through different levels, unlocks characters and upgrades them with equipment.• Dividing the screen in a coding part and a part that represents the player's moves helps children to learn, practice and improve their skills in programming (Kumar, 2014).• The game includes practice levels and hints for supporting students in comprehending programming concepts, as well as challenge levels for assessing the comprehension of programming concepts.• Available at: https://codecombat.com/
CheckiO	<ul style="list-style-type: none">• A <i>platform</i> that provides various tasks in order to support the learning process of the programming languages Python and JavaScript.• In contrast with the aforementioned games, CheckiO is targeted both at beginners and experienced programmers who want to advance their coding skills through <i>engaging challenges and fun tasks</i>.• Available at: https://checkio.org/
Empire of Code	<ul style="list-style-type: none">• A descendant of CheckiO with more game elements.• <i>Multiplayer online strategy game</i>.• Empire of Code gives the player more incentives since s/he has to gain points that will help her/him build a base, upgrade their buildings and develop units for strengthening the base, as well as protecting it against enemies and other players.• The game supports cooperation between players in order to organize their strategy against another base and destroy it.• Available at: https://py.checkio.org/blog/empire-code-space-strategy-game-checkio/

Analysis and Design of Py-rate Adventures

The serious game Py-rate Adventures aims to educate the player on the basic concepts of programming with Python. The theory of the game is embedded in such a way that the player acquires knowledge gradually with more complex programming concepts as s/he proceeds to the game. The game has *six levels* in total and each one of them covers different programming concepts in Python. In this section basic data about the game are presented and design decisions are briefly analyzed in the context of the *Educational Games Design Model* proposed by Ibrahim and Jaafar (2009).

Instructional Objectives

- Present, explain and familiarize the learner with basic programming concepts in Python through *brief and concise theory* and representative *examples* incorporated in the game. The educational content includes *variables*, manipulation of *strings*, decision making with *conditional statements*, *lists*, *tuples*, *dictionaries*, *loops* and *classes*.
- Enhance the process of learning programming through *interactive tests* during the game.

Game Objectives

Py-rate Adventures is a *2D platformer action game* in which the player controls an avatar and has to surpass several obstacles, avoid or destroy enemies and complete all the levels. Each level of the game introduces the player with new concepts, which will have to be assimilated by the player, in order to pass several obstacles in the tests that will appear.

Game Flow

- The flow of game is the same for each level, including an avatar character, a *pirate* controlled by the player in order to navigate through the level's map and gather all the treasure chests and avoid enemies (Figure 1).

- Each *treasure chest* contains theory and examples of the programming concepts that represent the learning goal of each specific level (Figure 2). When the player touches a treasure chest, the theory pops up on the map of the level, so that the player can review it.

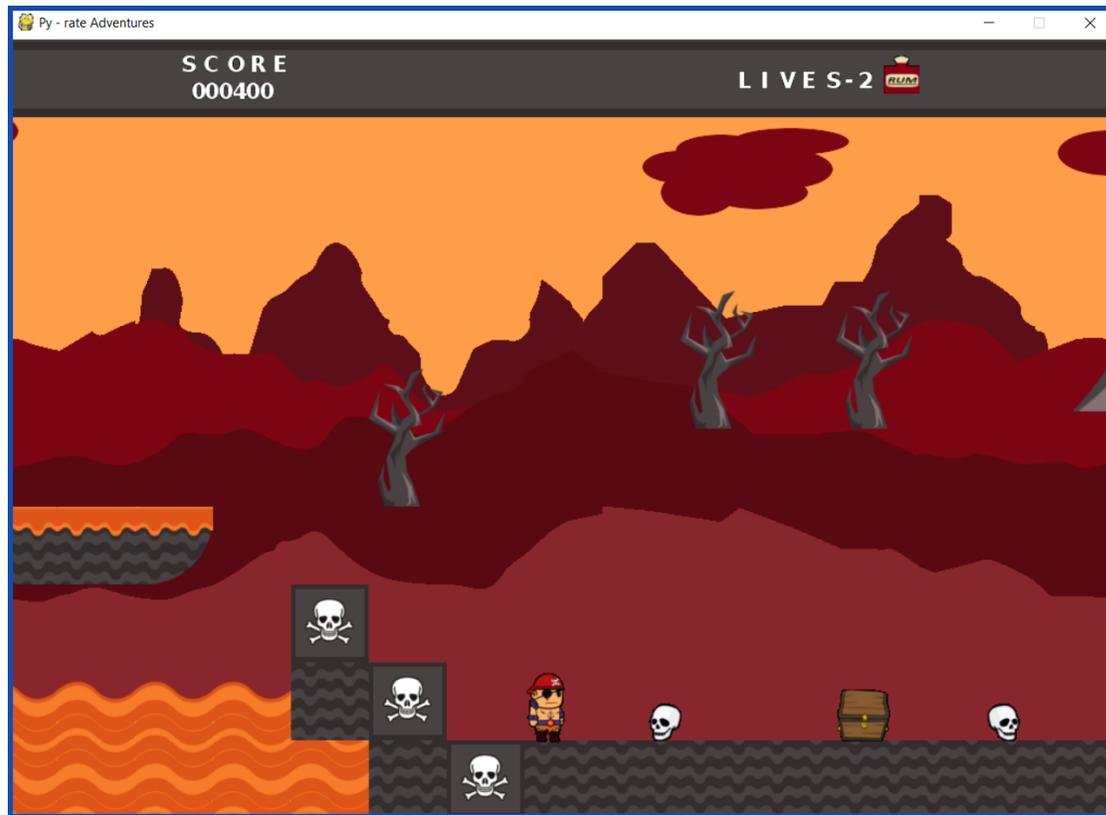


Figure 1. Player navigates through the map.

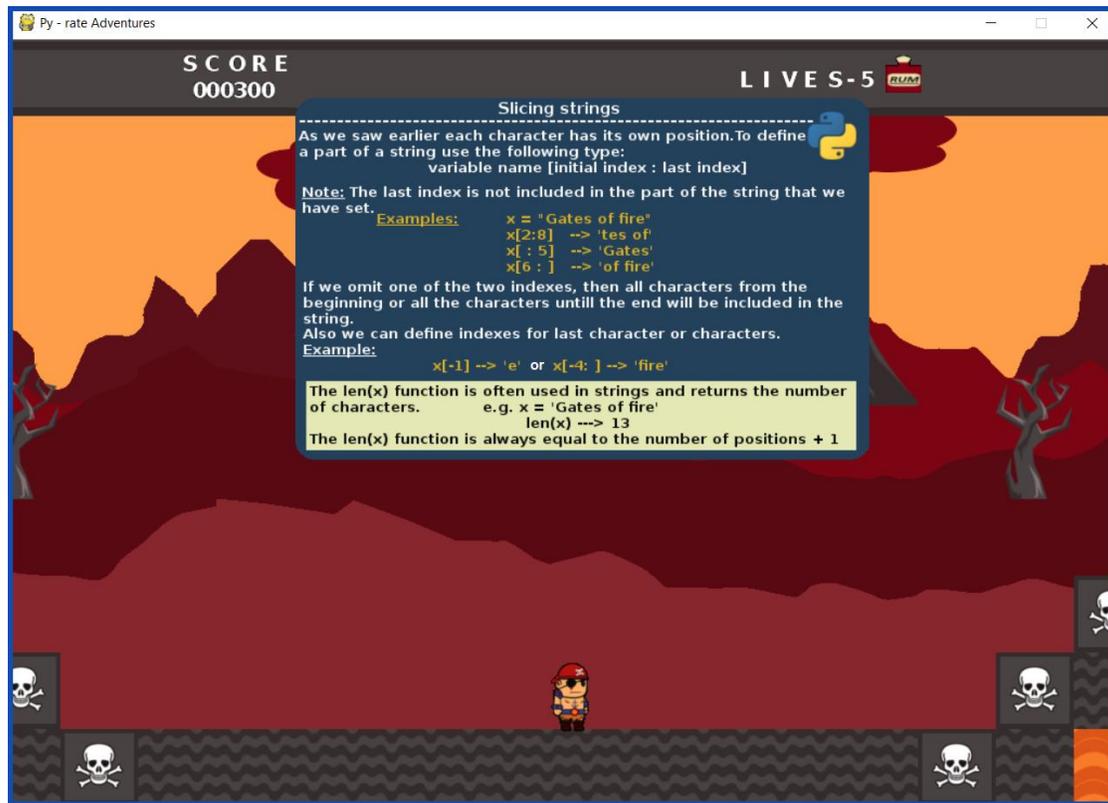


Figure 2. Treasure chests content - theory of the game and examples.

- As the player moves forward, *enemies* (skulls) appear in the level's map. Sometimes the player can just avoid them and move and sometimes the player has to jump and crush them improving the score (Figure 3). As the player progresses to the next levels, more enemies appear to raise the level of difficulty in the game.

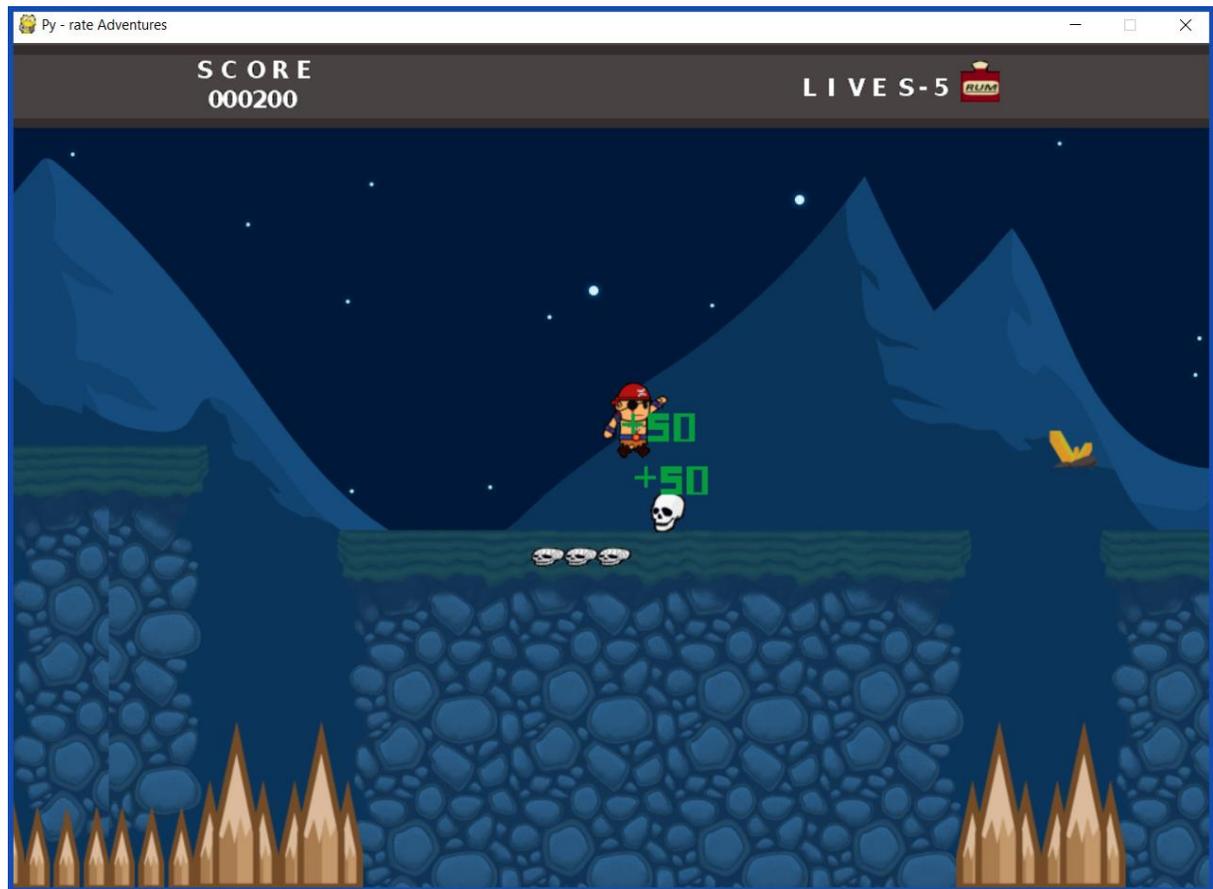


Figure 3. Player crushes the enemies and improves his/her score.

- The pirate is confronted with *obstacles* and the player has to answer correctly in several *quizzes* that are embedded in the map of each level in order to move on. The questions are based on the content of the treasure chests and aim to assist the player in comprehending the theory and testing their knowledge. These questions might be either theoretical or code-based (Figure 4). Specifically, the player has to define the execution result of a block of code or correctly select the parts of code that are missing.

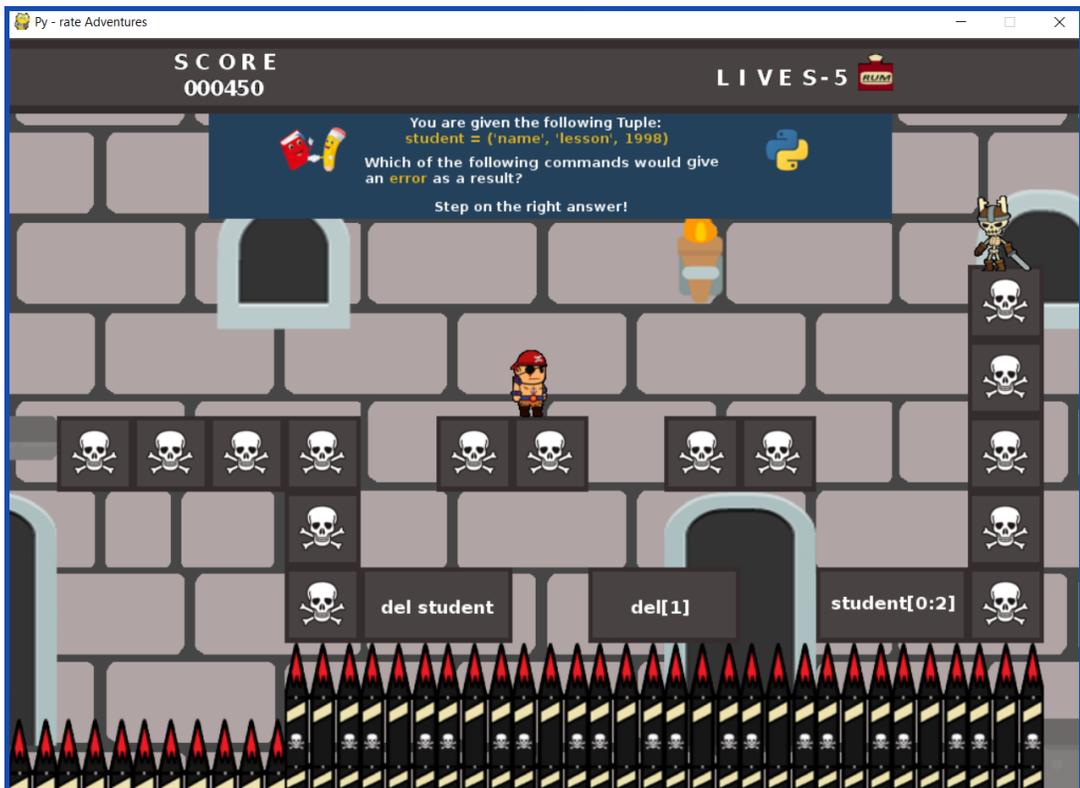


Figure 4. Quizzes for testing the player's knowledge during the game.

- After completing all the tests and surpassing several enemies and obstacles the player reaches the exit sign and completes the level. In the screen that appears the player has to decide whether s/he will move on to the next level or will play again the same level (Figure 5). The player has *five lives* in total to complete all the levels. The player loses a life by falling from a platform, falling to the spikes, being touched by the enemy or answering a question wrong. If the player loses all the lives the game starts again from the beginning of the level. Finally, the player has access to all six levels of the game and their learning goals and can select the level s/he wants to play.

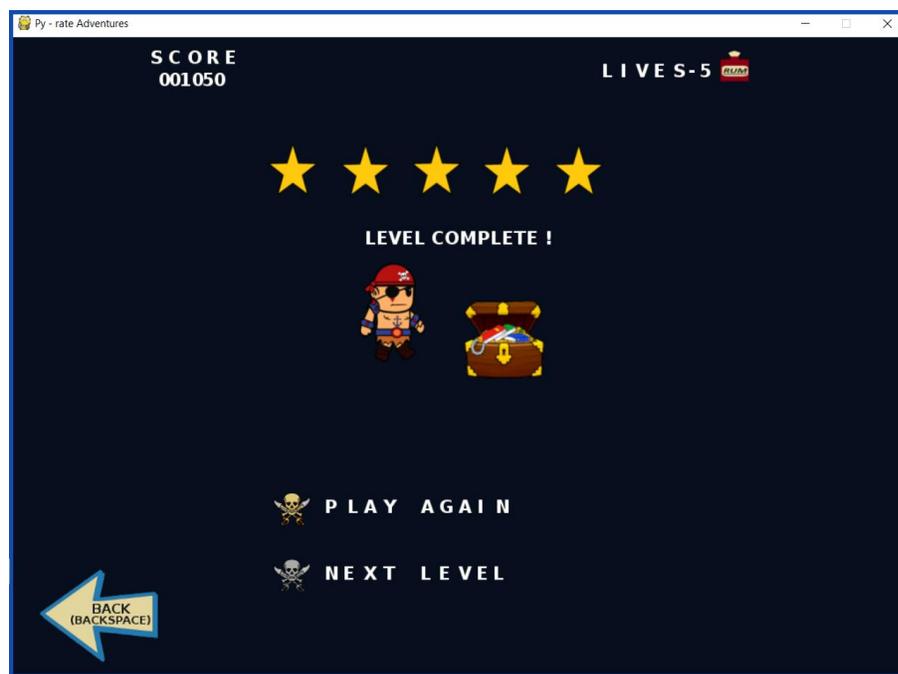


Figure 5. Level completion screen - the player can go to the next level, play again or return to the menu.

Target Audience

The game is aimed at those who want to make their first steps in programming or those who want to get familiar with the basic concepts of the object-oriented programming language Python. Moreover, the game does not require any prior programming knowledge and anyone interested can play it.

Preparation and Set-Up Time

Py-rate Adventures is a 2D platform keyboard game that has to be downloaded or copied to the player's computer. The game can be played on computers running Windows, by running the executable file that has been created for that purpose.

URL for downloading:

<https://sites.google.com/a/uom.edu.gr/stelios-xinogalos/serious-games/programming#TOC-Py-rate-Adventures:-a-2D-platform-game-for-learning-programming-in-Python>

Playing Time

The playing time varies from player to player as it depends heavily on her/his previous knowledge of programming, as well as the difficulty that s/he faces in comprehending the concepts that are introduced. The playing time also depends on the players' experience with this kind of games and how easily they adjust to the game rules. The average players' playing time is approximately 2-3 hours to complete all the levels of the game, but it depends on several factors as previously mentioned. Based on the pilot evaluation a player has to play the game 8 to 12 times on average in order to manage to complete all the levels, but exceptions may exist depending on several factors.

Game Design, Pedagogy & Learning Content Modelling

As already mentioned, Py-rate Adventures was designed based on the *Educational Games Design Model* (EGDM) proposed by Ibrahim and Jaafar (2009). This framework consists of three main axes: Game Design, Pedagogy and Learning Content Modelling. It is characterized as a design framework, which contains fewer elements than other well-known design frameworks, and consequently it is less complex and easier to understand (Ahmad et al., 2014). In Table 2 we present the way we approached the elements included in the *Game Design* axis of the EGDM model, in Table 3 the elements included in the *Pedagogy axis* and in Table 4 elements related to the *Learning Content Modelling* axis.

Table 2. Game Design in Py-rate Adventures.

Game Design	
Usability	
✓ Satisfaction	reminds the well-known platform game Super Mario; simple but attractive graphics; the player is rewarded for dealing with enemies; answering correctly in quizzes increases the score and the player's satisfaction
✓ Efficiency	obstacles (quizzes) are encompassed efficiently in the flow of the game; brief and concise theory and representative code examples are considered an efficient way of introducing novices to programming concepts; immediate feedback to the player's response to quizzes; ability to replay any level for succeeding better results
✓ Effectiveness	the game's efficiency contributes to its effectiveness as well; the player is tested for his/her comprehension of programming knowledge and receives immediate feedback that helps him/her realize effectively the correctness of his/her conceptions and dealing with misconceptions
Multimodal	
✓ Multimedia	the game includes graphics, sounds for notifying important events, text and source code appropriately highlighted with characteristic colors, basic animation that is appropriate for the selected game genre
✓ Interaction	the player interacts with enemies, treasure chests and obstacles; quizzes are interactive providing immediate feedback interleaved with the scenario of the game (the pirate falls from the platform in case of a wrong answer)
Fun – Challenge	
✓ Clear goals	both the learning and game goals are straightforward and clearly presented to the player
✓ Uncertain outcome	the uncertain outcome is not a main feature of platform games, however there is a variety in the number and movement of the enemies, the type of the obstacles and quizzes through the levels
✓ Self esteem	there is the ability to replay a level in order for the player to increase his/her score and improve his/her performance, so as to gain self esteem both as player and as a learner

Table 3. Pedagogy issues in Py-rate Adventures.

Pedagogy	
Learning Outcomes	the game has clear learning outcomes (see instructional objectives) regarding the introduction to programming concepts with Python; the game aims mainly at the first two levels of Bloom’s taxonomy of learning outcomes, namely <i>knowledge</i> and <i>comprehension</i> (treasure chests), and at a lesser degree at the third level of <i>application</i> (obstacles-quizzes)
Motivation Theory	learning a cognitively demanding subject, such as programming, through a game is considered highly motivating; self-learning and self assessment in the player’s own pace; the player learns through his/her mistakes in an environment controlled by him/her with no consequences or exposure to a teacher/class that can be stressful
Self-learning	the game does not require previous programming knowledge; it contains the necessary theory and examples, which are presented clearly and concisely; it contains a self-assessment module with immediate feedback; the player is free to replay the levels s/he wants in her/his own pace
Problem-solving	due to the area of the game problem solving is inherent

Table 4. Learning Content Modelling in Py-rate Adventures.

Learning Content Modelling	
Syllabus Matching	the syllabus was selected in order to be appropriate for people with no prior programming experience; it is the typical syllabus of a seminar/short course on introduction with Python; it is appropriate for self-learning
Scaffolding	the game includes brief and concise units of theory and examples; programming concepts are presented progressively within a level and through levels; the player can study the material in his/her own pace; quizzes are used for self-assessment; the game proposes the sequence of levels but each level can be replayed

Implementation of Py-rate Adventures

Py-rate Adventures was implemented using *Python 3*, due to the simplicity of code written in Python, its support of object oriented programming and the various useful modules it provides. In order to expedite the development of the game the free and open source Python library of Pygame was used. *Pygame* was implemented in the prototypes of the SDL library (Simple DirectMedia Layer) and was first released in 2000. Pygame provides a significant amount of modules that help a lot in management and processing the objects of the game. The most

important modules of Pygame that were used and made easier the implementation of Py-rate Adventures are presented in Table 5.

Table 5. Modules of Pygame utilized in the development Py-rate Adventures.

Module name	Description
Pygame.display	Displays the window of the game on the screen
Pygame.draw – pygame.blit	Designs the graphic objects of the game and displays them on the screen
Pygame.surface	Manages the interface for all graphics and images
Pygame.image	Loads and saves images
Pygame.sprite	Manages the animation of the graphic objects and their collisions
Pygame.rect	Manages the shape of the graphic objects
Pygame.transform	Moves and transforms the shape of images
Pygame.mixer	Loads and plays sounds
Pygame.time	Manages the frames per second (fps) in the game
Pygame.event	Manages the actions of the user/player (keyboard, mouse)
Pygame.key	Recognizes the key that the player presses from the keyboard

The IDE that was used for the development of Py-rate Adventures was *Pycharm* (<https://www.jetbrains.com/pycharm/>). Overall, the game consists of 75 classes and approximately 8000 lines of code. In addition, the game has been developed in such a way that is possible to add another language for the graphical user interface (besides English and Greek currently supported) and embed the appropriate educational material without much effort.

The six levels of the game were designed using the *Tiled – Map Editor* (<https://www.mapeditor.org/>). Free graphic assets (www.itch.io, www.gameart2d.com, www.opengameart.org) and sound effects (www.noiseforfun.com, www.freesound.org, www.zapsplat.com, www.soundboard.com) were utilized.

Pilot evaluation of Py-rate Adventures

The pilot evaluation of Py-rate Adventures was carried out through an anonymous questionnaire that was completed by the participants, right after their hands-on experience with the game. In this section we present the methodology, the demographic data of the participants and the results of the evaluation.

Methodology

The questionnaire used for the evaluation of Py-rate Adventures was based on the MEEGA+ framework. This framework was designed as a tool for evaluating educational games in terms of the players' experience and perceived short-term learning (Petri et al., 2016).

The player experience axis is split into several dimensions: usability, confidence, challenge, satisfaction, social interaction, fun, focused attention and relevance. The axis of perceived learning refers to the learning outcomes of the game, and more specifically it refers to short-term learning, learning goals and game-specific learning goals that can vary a lot.

The final questionnaire that was designed consisted of the following three sections containing 39 questions in total:

- demographics (4 questions),
- player experience (29 questions),
- perceived learning outcomes (6 questions)

The closed-ended 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 percentages for each possible response in each question were calculated.

Participants

The participants were graduates and postgraduate students of an Interdepartmental Programme of Postgraduate Studies in Information Systems. Thirty-one participants responded in an invitation sent by email to play and evaluate Py-rate Adventures in a time period of ten days. The participants anonymously filled in the on-line questionnaire prepared using Google forms. This research has been conducted following the ethical requirements established by Greece national board of ethics.

Results and discussion

Demographic information. Some basic demographic information for the 31 participants is presented in Figure 6. The participants had different skills and knowledge in programming due to the variety in their bachelor studies (economics, business administration, informatics etc). It is worth noting that 45% of the participants had little or no previous knowledge in the programming language Python, while nearly 40% of them rarely or never play digital games.

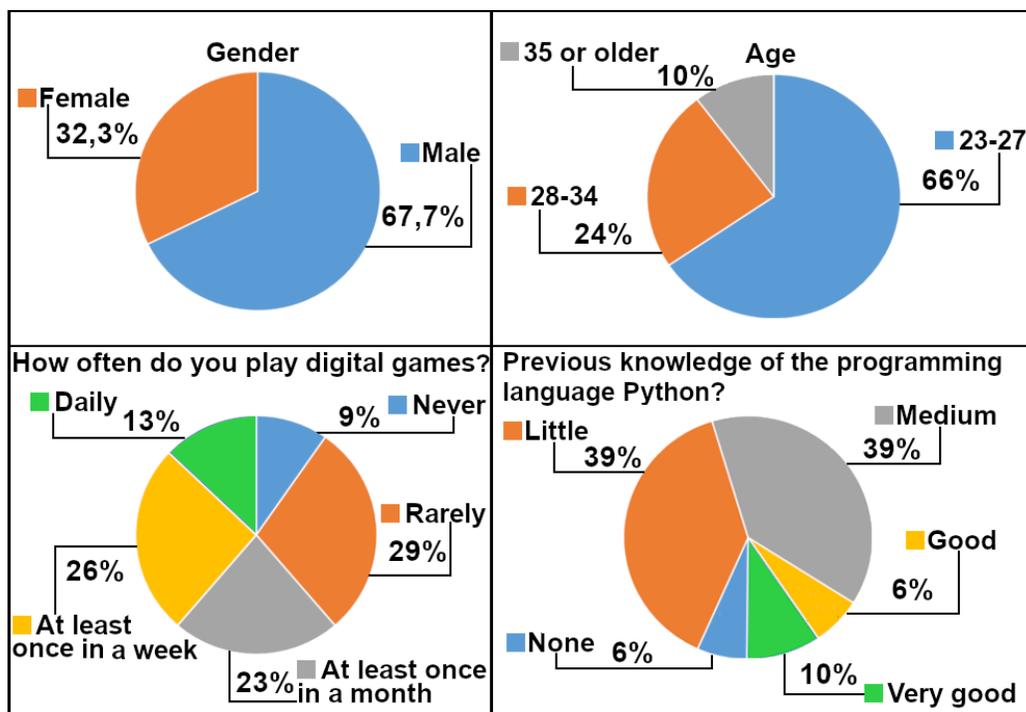


Figure 6. Demographic data of the pilot evaluation

Usability. The results of the evaluation regarding the usability of Py-rate Adventures are presented in Figure 7.1. The results of the evaluation highlight the attractive design of the game. *Game design, fonts, and colors* were positively evaluated. More specifically, 81% of the participants agree or strongly agree that the design is attractive, while 90% of them agree or strongly agree that the fonts of the game are legible. Moreover, 87% of the participants agree or strongly agree that most people would *learn to play the game quickly*, while 97% of them agree or strongly agree that the *rules of the game are clear and easy to understand*. Finally, it is worth mentioning that 84% of the participants disagree or strongly disagree that the game requires previous knowledge for someone to play it.

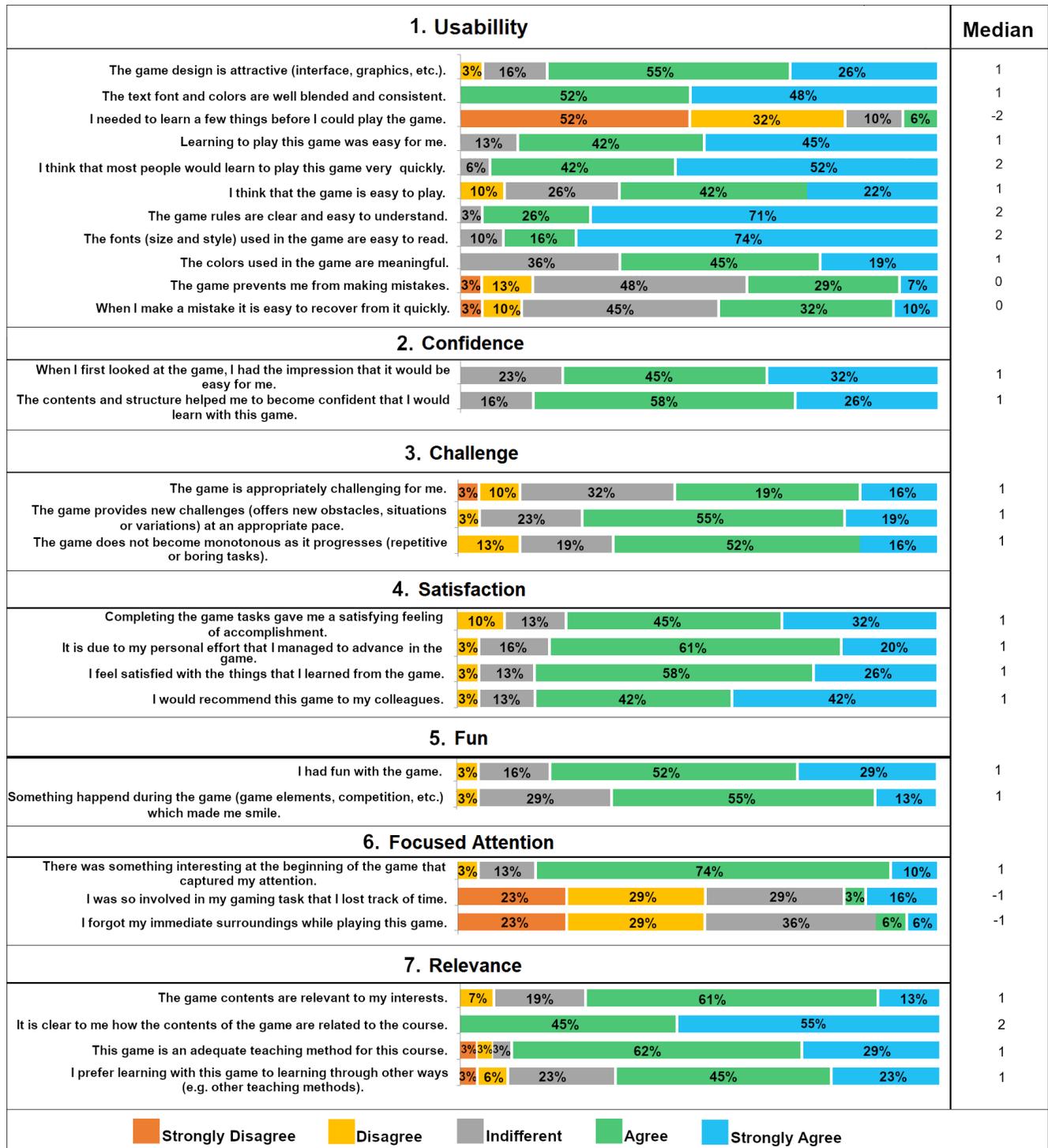


Figure 7. Evaluation results on player experience

Confidence. The results of the evaluation regarding the confidence of the participants are presented in Figure 7.2. More than three out of four participants agree or strongly agree that the *first impression* of the game as well as its *content and structure* reinforce a player's self-confidence.

Challenge. Figure 7.3 presents the results regarding the challenging aspects of the game. More than half of the participants (39% agree and 16% strongly agree) believe that the game was *appropriately challenging* for them. In addition, most of the users (55% agree and 19% strongly agree) agree that the game *provides new challenges* and also *it is not monotonous and boring* (52% agree and 16% strongly agree). According to the results of this category, the game provides the appropriate challenges for a player by giving him/her more incentives to continue playing, as all questions have a median of 1 (agree).

Satisfaction. The results that refer to user's satisfaction during game play are presented in Figure 7.4. The game was positively evaluated by the participants, achieving a high percentage in all the questions of this dimension (median 1 - agree). The highest positive percentage (84%) was recorded for the questions related to the player's satisfaction with what s/he learned from the game and whether s/he would recommend the game to her or his colleagues.

Fun. The results regarding the fun that the game provided to the participants are presented in Figure 7.5. Four out of five (52% agree and 29% strongly agree) participants stated that they had fun while playing the game, while something happened during gameplay that made them smile (55% agree and 13% strongly agree).

Focused attention. The results that concern the focused attention of the player while playing the game are presented in Figure 7.6. The game managed to attract the players' interest and draw their attention (74% agree and 10% strongly agree), but not to such a degree that will make them lose the sense of time during game play (23% strongly disagree and 29% disagree). Based on these results the game has room for improvement on this category by adding some elements that will achieve a greater degree of focus from a player during game play.

Relevance. Figure 7.7 presents the participants' perceptions on the relevance of the game with programming. From the participants' responses, it is clear that the game is *relevant* with programming (median = 2 – strongly agree) and that the users *prefer this teaching method* comparing to others (45% agree and 23% strongly agree). Finally, it is worth noting that all the questions that are related to the relevance of the game with programming have recorded positive answers (median 1 or 2 – agree or strongly agree), which confirms even more the purpose of implementing this educational game.

Perceived learning. In Figure 8 the results of the evaluation that refer to the educational objectives of the game and the perceived learning of the users after their game play experience are presented. The vast majority of the users agree that the game helped them to understand basic concepts of programming in Python and support that they learned something new by playing Py-rate Adventures (median = 1). Furthermore, all the questions that concern the category of perceived learning have recorded a median of 1 (agree), which confirms to a large extent that the educational objectives of this game have been achieved.

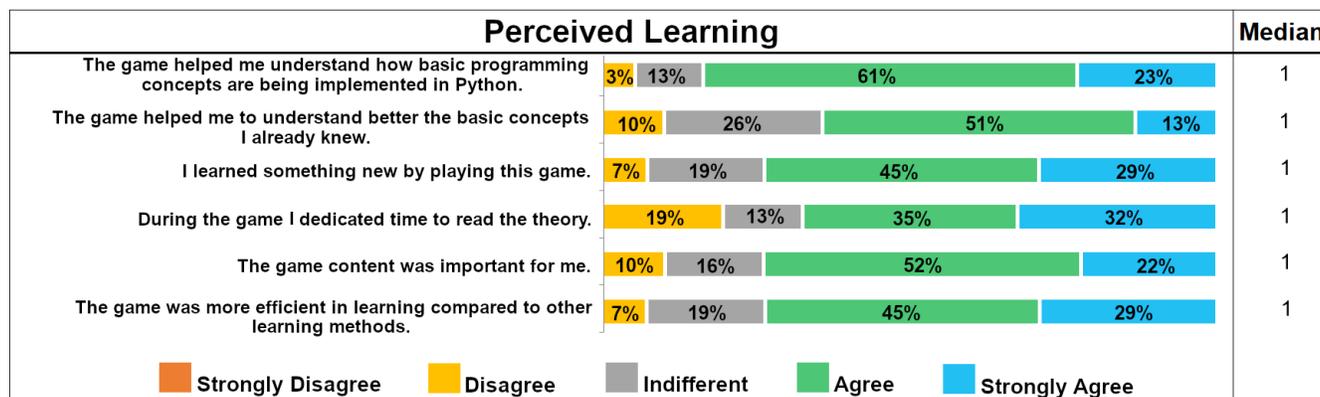


Figure 8. Evaluation results on perceived learning

Conclusion

Py-rate Adventures is a 2D platform serious game for introducing novices to basic programming concepts using Python. The pilot evaluation of the game gave positive results both in terms of player experience and short-term learning, as they are defined in the MEEGA+ model (Petri et al., 2016). Regarding the *players' experiences*, all the categories achieved high percentages of positive responses for most of the questions. The category with the less positive results was the category of *focused attention*, which could be improved in order to achieve greater immersion of the player during game play. Regarding the *perceived short-term learning*, the results of all the questions were rather positive and encouraging. Py-rate Adventures gave the participants the chance to comprehend programming concepts in Python or to comprehend better concepts they already knew, while the majority of the participants stated that they prefer learning through the game compared to other learning approaches. Of course, it is considered necessary for this serious game to be evaluated by students of secondary

education, in order to extract stronger conclusions about its value as an auxiliary tool to learn basic programming concepts with Python.

The results of the pilot evaluation of Py-rate Adventures are analogous to those of jAVANT-GARDE (Galgouranas & Xinogalos, 2018) which is a platform game for an introduction to programming using Java. jAVANT-GARDE was evaluated by secondary education students using the MEEGA+ model as well. The positive results of the pilot evaluation of both games provide strong indications that platform games are appropriate for introducing students to programming concepts no matter what the underlying programming language is. Both games incorporate brief and concise theory and examples, while students assess their knowledge through multiple choice questions, filling in gaps or defining the execution results of excerpts of source code. These types of activities provide the opportunity to study and experiment with programming concepts prior implementing programs in a typical programming environment. However, activities that require the development and testing of complete programs in the context of the game would give the chance to utilize the game in the whole process of teaching and learning programming. We consider it important to extend the aforementioned games in this direction and study their impact on learning programming.

Declaration of Conflicting Interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

References

- Ahmad, M., Rahim, L. A. & Arshad, N. I. (2014). A review of educational games design frameworks: An analysis from software engineering. *International Conference on Computer and Information Sciences (ICCOINS)*, Kuala Lumpur, Malaysia. 3-5 June 2014. DOI: [10.1109/ICCOINS.2014.6868452](https://doi.org/10.1109/ICCOINS.2014.6868452)
- Combefis, S., Beresnevičius, G., & Dagienė, V. (2016). Learning programming through games and contests: overview, characterisation and discussion. *Olympiads in Informatics*, 10(1), 39-60.
- Galgouranas, S., & Xinogalos, S. (2018). jAVANT-GARDE: A Cross-Platform Serious Game for an Introduction to Programming with Java. *Simulation & Gaming*. Vol. 49, Issue 6, 751-767.
- Giannakoulas, A., Xinogalos, S. (2018). Current Trends in On-line Games for Teaching Programming Concepts to Primary School Students. *Proceedings of the International Conference on Technology and Innovation in Learning, Teaching and Education (TECH-EDU 2018)*, Thessaloniki 20-22 June 2018 (in press).
- Ibrahim, R., & Jaafar, A. (2009). Educational games (EG) design framework: Combination of game design, pedagogy and content modeling. *International Conference on Electrical Engineering and Informatics* (pp. 293-298). Selangor, Malaysia. doi:10.1109/ICEEI.2009.5254771
- Kumar, D. (2014). Digital playgrounds for early computing education. *ACM Inroads*, 5(1), 20-21. DOI: <https://doi.org/10.1145/2568195.2568200>
- 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*, Vol. 10, Issue 2, 219-235, doi:10.1109/TLT.2016.2556666.
- Petri, G., von Wangenheim, C. G., & Borgatto, A. F. (2016). MEEGA+: An evolution of a model for the evaluation of educational games (Technical report, INCoD/GQS.03.2016.E). Brazilian Institute for Digital Convergence.
- Stege, L., Van Lankveld, G., & Spronck, P. (2011). Serious games in education. *International Journal of Computer Science in Sport*, 10(1), 1-9.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.