

jAVANT-GARDE: A Cross-Platform Serious Game for an Introduction to Programming with Java

Abstract

Background. Programming is widely known to be a cognitively demanding field. **Learning programming** can be a strenuous task without the proper guidance and methodology. Students often face difficulties in understanding even the most basic and fundamental concepts of programming, leading to their confusion and disappointment. Serious games for programming have the potential to alleviate students' difficulties with programming.

Aim. This article introduces a new **serious game** named jAVANT-GARDE which aims to educate the player on the **basic concepts of programming** and object-oriented programming through Java. By design, the game is suitable for everyone, even for players with no previous knowledge of programming. This article aims to present important information regarding the design, implementation and **pilot evaluation** of jAVANT-GARDE.

Method. The game was evaluated by 42 high school students (age 16-17) in terms of **perceived player experience** and short-term **learning**. The participants played the game for one hour in the school lab and then answered a questionnaire based on the MEEGA+ framework.

Results. Students evaluated positively the game in terms of its usability, confidence in playing the game, challenge, satisfaction, fun, focused attention and relevance. Moreover, the majority of students stated that would prefer learning programming with this game to learning through other teaching methods.

Conclusions. The results of the pilot study and students' replies in the questionnaire are rather positive. This gives us the initiative to proceed with studying the effect of the game in long-term learning of programming concepts.

Keywords

Serious games, educational games, learning programming, evaluation

Basic Data

Instructional objectives

- Present, explain and familiarize students with basic concepts of programming, including variables, methods, loops, conditionals, arrays, and classes.
- Facilitate the process of learning programming through interactive tasks.

Game objectives

The game is a 2D platformer on which the player controls a robot and guides it through a series of levels.

All the levels contain specific learning goals and, as the game progresses, the player is introduced to new concepts, which he/she has to put into practice. The basic structure of a level is the following:

- The player controls a robot to navigate through the level's map in order to find info signs that contain the theory of that particular level.
- After studying and comprehending the theory, the player stumbles upon quizzes and various quests.
- To complete the quests, the player is asked to write code, fill in blanks, or put pieces of code in the correct order.
- Eventually, when the player completes all the quests, the path to the termination portal is cleared and the player moves on to the next level.

Debriefing format

The game includes integrated self-assessment modules aiming to provide a reflection-in action process (Clapper, 2018). Specifically, students reflect on the experience gained through playing and studying info signs by auto-evaluating their understanding of the aforementioned experience and knowledge through

carrying out the various integrated programming activities that give immediate feedback. Students can take the time needed to reflect on their experience, while iterations are possible in order to achieve proper debriefing (Crookall, 2014). This means that students can go back to study the info signs and repeat the activities, as long as they feel that they have comprehended the material. Of course, if they lose all their health points they have to repeat the level!

Target audience

The game does not require any previous programming knowledge and anyone interested in it may benefit from the game.

Preparation and set-up time

The desktop version of the game has to be downloaded or copied to the participants' computers and Java must be installed as well.

Playing time

The playing time varies from player to player as it depends heavily on his/her previous knowledge of programming, as well as the difficulty that he/she faces in comprehending the concepts that are introduced. The average player's playing time is approximately 2-4 hours to complete all the levels of the game, but it depends on several factors as already mentioned. However, the game saves players' progress and can be played on each one's own pace.

Participation materials required

The game can be played on computers running Windows, Mac, or Linux with Java installed and on Android, iOS and Blackberry mobile devices. It can also be played online through a web browser. The essential educational material is incorporated in the game and no additional information is required.

Introduction

Throughout the last decades, video games have secured a place at the apex, among the most thriving forms of entertainment. They are embellished with such diversity that they appeal to people of all ages and different tastes. This fact has led to various studies concerning the use of video games in fields other than just entertainment, taking into account their application on the educational domain as well. Game-based learning and serious games have come to be widely used for educational purposes, providing a way to integrate the element of entertainment and other traits of video games into the learning process. Studies have shown that under proper design and methodology, serious games can be more effective than traditional or other learning techniques.

Programming is among the plethora of subjects that have been utilized in serious games. It is widely known that learning programming is accompanied with many difficulties, while several misconceptions arise when students are introduced to programming. When it comes to object-oriented programming students find it difficult to comprehend even the main concepts of “objects” and “classes” (Xinogalos, 2015). The serious game jAVANT-GARDE, presented in this article, was developed as an attempt to motivate people to get involved with programming and to support them in understanding, even its most complicated concepts (such as “objects” and “classes”). Playing the levels of the game progressively provides a smooth transition between these concepts to the player, in an attempt to disambiguate and clarify any misconception. The game is designed specifically for people with no previous knowledge of programming, as well as novices. One distinguishing feature of jAVANT-GARDE is the fact that it is a cross-platform serious game that runs both on computers and mobile devices. Moreover, the game is open source and can be utilized and even extended by anyone interested.

In this article, we present the serious game jAVANT-GARDE along with its design which is based on the *Educational games design model* proposed by Ibrahim and Jaafar (2009). Additionally, we analyze key elements regarding the implementation of the game, such as the programming framework that was

utilized for its development, namely the LibGDX framework. A detailed description of the evaluation regarding its methodology, participants and results is also presented. Finally, we conclude and suggest plans for future research.

Analysis and Design of jAVANT-GARDE

The serious game jAVANT-GARDE aims to educate the player on the basic concepts of programming with Java. The theory of the game is organized into categories and levels, in order that the learning goals are distributed in a clear and structured manner. 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 main logic of the levels adheres to the following structure:

- The player studies the theory of the level from signs that are placed throughout the map called *Info Signs*. Each Info Sign contains a part of the theory corresponding to each level's learning goals.
- The path of the robot is blocked and the player has to answer quiz questions in order to move on. The questions are based on the content of the Info Signs and aim to assist the player in understanding the theory.
- Finally, the player has to put what he/she learned into practice and complete various quests. To do so, he/she has to write code, fill in blanks, and put pieces of code in the correct order.

After completing all the quests, the player's path is no longer blocked and he/she is free to reach the termination portal which ends the level and unlocks the next one. If the player loses all of his/her health points either by answering wrong or falling from the platforms, he/she has to repeat the level.

It has been challenging for serious game developers to successfully convey the educational material of their games to the players, given that it is a relatively new field and there are not many well-established design guidelines (de Freitas, 2006; Ibrahim & Jaafar, 2009). A serious game has to be embedded

effectively into practice and in accordance with sound pedagogic principles and design, in order to be considered an effective learning tool (de Freitas, 2006). Therefore, the design phase of jAVANT-GARDE was taken into great consideration. After carefully analyzing serious game design frameworks, it was decided that the *Educational games design model* proposed by Ibrahim and Jaafar (2009) was appropriate for the design of this game. The main factor that led to this decision is that this model is designed specifically for self-learning games with integrated self-assessment modules (Ibrahim & Jaafar, 2009). Additionally, this model is a product of research of previously established educational game design models and its clear structure provides well-defined guidelines.

Ibrahim and Jaafar's (2009) model consists of three main axes; *game design*, *pedagogy* and *learning content modelling*. These axes mainly focus on usability, multimodality, fun, problem solving and syllabus matching. Tables 1 to 3 present the design decisions taken for each axis of the model regarding jAVANT-GARDE.

The *game design* axis consists of three elements: usability, multimodality and fun (Table 1). Usability is measured in terms of satisfaction, efficiency and effectiveness. Multimodality underlines the need to properly combine multimedia (e.g. text, graphic, audio, video, animation) with interactivity so that the players can interact with the game and receive immediate feedback. Fun is the final element and it is defined by the clear goals that the game sets, the uncertain outcome that intrigues the players, and the boost of the players' self-esteem while playing the game and facing the presented challenges.

Table 1. Game Design.

| Factor (Ibrahim & Jaafar, 2009) | Design decision |
|---|--|
| Usability - Satisfaction (de Freitas & Jarvis, 2006; Malliarakis, Satratzemi, & Xinogalos, 2014; | The interactive game environment, the smooth animations, the appealing graphics, the visualization of the in-game paradigms, the joy of completing quests and exploring the game's levels are all factors that contribute to the player's personal satisfaction. Furthermore, the player has the ability to play the game in the |

| | |
|--------------------------|---|
| Salen & Zimmerman, 2004) | platform that satisfies him/her more (desktop, web, mobile). |
| - Efficiency | The clear goals and the laconic theory aim to maximize the efficiency of the game. All the keywords in the texts and the examples are colored appropriately to make sure that the crucial parts of the theory stand out and are easier for the player to understand and retain. |
| - Effectiveness | Moreover, the game offers a variety of interactive tasks for the evaluation of the player's learning progress including writing code, filling the gaps, putting pieces of code in the correct order and answering quizzes. The consistency of each consecutive level and its learning goals in conjunction with the clarity of the theory renders the game suitable for people with little or even no knowledge of programming at all. |

Multimodal

| | |
|---|---|
| - multimedia and interaction (de Freitas & Jarvis, 2006; Malliarakis et al., 2014; Salen & Zimmerman, 2004; Yusoff, Crowder, Gilbert, & Wills, 2009) | The game aims to hone the player's experience by utilizing various multimodal objects like appealing graphics and animations, meaningful texts, arousing music and sound effects, and also by harmonizing these objects according to the player's actions in order to achieve a highly interactive environment. Actions like jumping, colliding, collecting items, using items, giving a correct/incorrect answer, completing a level, etc., are supported by audiovisual effects. |
|---|---|

Fun – Challenge

| | |
|---|---|
| - clear goals (de Freitas & Jarvis, 2006; Malliarakis et al., 2014; Salen & Zimmerman, 2004) | All the levels share a common ultimate goal: The player has to reach the portal in order to complete the level and move on to the next one. At any given point, the player has a particular well-defined quest or task to complete. For additional clarifications, the robot-announcer guides the player throughout the levels. |
| - uncertain outcome (Malliarakis et al., 2014) | As the player progresses through the levels, he/she explores new challenges and aspects of the game which intend to arouse his/her curiosity. Some of those aspects are the various visualizations of the code that the player writes, the outcome of the player's actions and answers, and also the riddle-resembling nature of the game in general. |
| - self esteem | Each level of the game comprises of various individual tasks and quests which players are called to complete. These challenges, simple or subtle, are all based on the theoretical content of that particular level and can be completed by the player without previous knowledge of programming. |

Pedagogy is the axis that incorporates the educational aspect in the design (Table 2). Learning outcomes is one of the elements included in the axis which emphasizes on the first three levels of Bloom's

Taxonomy - knowledge, comprehension and application. Furthermore, the model promotes self-learning using integrated self-assessment modules. By this means, it allows students to teach themselves by playing and also to assess their own performance, while encouraging them to study at their own pace. Moreover, this model highlights the need for the game to inspire motivation to the players and also, to promote high order thinking skills by adapting problem solving to the game content.

Table 2. Pedagogy.

| Factor (Ibrahim & Jaafar, 2009) | Design decision |
|--|---|
| Learning Outcomes (de Freitas & Jarvis, 2006; Malliarakis, Satratzemi, & Xinogalos, 2014; Salen & Zimmerman, 2004; Yusoff, Crowder, Gilbert, & Wills, 2009) | The game aims to introduce the basic concepts of programming to the player using Java. These concepts include variables, conditionals, loops, methods, arrays and classes. The structure of all the levels of the game follows a knowledge-comprehension-application sequence (Figure 1); a sequence that is based on the first three levels of Bloom's Taxonomy. At the beginning of each level the player is advised to study a specific part of the level's theory. Afterwards, he/she is called to answer quiz questions in order to get a better understanding of the theory. Finally, the player has to utilize what he/she learned to complete various quests by writing code, putting lines of code in the correct order, filling blanks, and much more. |
| Motivation Theory (de Freitas & Jarvis, 2006; Malliarakis et al., 2014) | As the game progresses and the theory expands new methods of evaluation are introduced in order to capture the player's interest and motivate them to continue playing and learning. All the levels of the game except for the first one are initially locked and the player must complete them in succession in order to move on. This design decision intends to boost the player's will to finish the game by unlocking all of its levels. |
| Self-learning | The division of the theory into organized categories and levels provides the players with the opportunity to sort out the information they receive. Moreover, players can choose to revise any previously completed level to further familiarize themselves with its theory and study at their own pace. |
| Problem Solving (Malliarakis et al., 2014) | Each time the player is prompted to study a part of the theory, he/she is expected to put the theory into practice and solve some problems in the form of quests and quizzes. On some occasions, the player has to study a simple example from the theory and then follow its reasoning to get to the desired solution of a problem. |



Figure 1. Basic structure of a level in jAVANT-GARDE.

As far as *learning content modelling* is concerned, it is stressed that the syllabus must match the intended learning outcomes in order to achieve the learning goals of the game (Table 3). Finally, the learning content of the game must be organized and offered progressively to the player in order to ensure

scaffolding, allowing the player to process and assimilate all the information and knowledge that the game aspires to impart.

Table 3. Learning content modeling.

| Factor (Ibrahim & Jaafar, 2009) | Design decision |
|---|--|
| Syllabus Matching (Malliarakis, Satratzemi, & Xinogalos, 2014; Salen & Zimmerman, 2004; Yusoff, Crowder, Gilbert, & Wills, 2009) | The main source of information regarding the theoretical framework of the game is the Java tutorials from Oracle's website: https://docs.oracle.com/javase/tutorial/ . Moreover, several introductory programming courses were studied in order to validate syllabus matching. |
| Scaffolding (Malliarakis et al., 2014; Yusoff et al., 2009) | The syllabus of the game is organized in small comprehensive units which are introduced to the player progressively. In this way, even players with no previous experience in programming are able to understand the theory and cope with the challenges of the game. Additionally, players may repeat any level that they previously completed in order to revise its theory and construct in a step-by-step manner their knowledge on programming. |

Implementation of jAVANT-GARDE

Implementing the game required a great amount of research and decision making, especially in the early stages of development. One of the most crucial steps in those 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 LibGDX framework was the most appropriate one for the implementation of jAVANT-GARDE. LibGDX is an open-source Java framework that provides a powerful set of tools to the developer and allows flexible development without enforcing strict methodology. It supports cross-platform development in a variety of operating systems including Windows, Mac, Linux, Android, BlackBerry, iOS and the Web. The features packed in the LibGDX framework along with other supplementary open-source software were utilized for the

development of jAVANT-GARDE. Two additional major factors that led to the decision to use this framework are its neat documentation and its immense and active community.

The following supplementary open-source software aided in the development of jAVANT-GARDE:

- **Tiled – Map Editor** (<https://www.mapeditor.org/>). Software used for designing 2D tile maps. LibGDX is compatible with the files generated by Tiled and they can be accessed and utilized programmatically.
- **Skin Composer** (<https://ray3k.wordpress.com/software/skin-composer-for-libgdx/>). Skin Composer is used for editing LibGDX skins. Skins define the look-and-feel of the User Interface (UI) of the LibGDX apps.
- **Hiero – Bitmap Font Tool** (<https://libgdx.badlogicgames.com/tools.html>). Hiero is a bitmap font packing tool. It generates fonts in the format that LibGDX works with.
- **LibGDX – TexturePacker** (<https://libgdx.badlogicgames.com/tools.html>). TexturePacker packs images in a way that is convenient for LibGDX in order to minimize resource use and boost performance.

The IDE that was used for the development of the serious game was **Android Studio** (<https://developer.android.com/studio/>). Overall, the game consists of 55 classes and approximately 8000 lines of code.

Another stage of implementation, which was also proven to be time-consuming, was the search for suitable and attractive assets. Therefore, the audio and most of the graphics of the game were carefully selected among a large quantity of free assets that are available on the web. Furthermore, some of the graphics were created by the first author of this article. A lot of effort was put into blending the assets together, while some editing was required as well. The graphics of the game were taken from three specific websites (<http://game-icons.net/>, <https://www.glitchthegame.com/>,

<https://www.gameart2d.com/freebies.html>). As far as the game's audio is concerned, it is classified into music (<https://www.jewelbeat.com/free/>) and sound effects (<http://www.noiseforfun.com/>).

Pilot Evaluation of jAVANT-GARDE

The pilot evaluation of jAVANT-GARDE was carried through a questionnaire that was filled in by the participants, right after their hands-on experience with the game. In this section we present the methodology, the demographic data regarding the participants along with the results of the evaluation.

Methodology

First and foremost, a questionnaire had to be prepared for the evaluation of jAVANT-GARDE. The questionnaire that was prepared was based on the MEEGA+ framework (Petri, von Wangenheim & Borgatto, 2016). MEEGA+ is a framework that was created for the purpose of evaluating educational games' perceived quality with respect to the player experience and perceived learning from the point of view of students in the context of undergraduate computing courses (Petri et al., 2016). The player experience quality factor is fragmented into several dimensions: usability, confidence, challenge, satisfaction, social interaction, fun, focused attention and relevance.

The final questionnaire consisted of a demographic data section, a quality evaluation section with 32 questions regarding the game experience and perceived learning of the students, and also a section for comments and feedback. The evaluation 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). The questionnaire was filled in by the participants, right after they had played the game in the school's computer lab. Utilizing the proposed MEEGA+ model, the median and percentages for each possible response in each question were calculated.

Participants

The pilot evaluation took place in a public Secondary Education School in Thessaloniki, Greece. The game was evaluated by 42 high school students (age 16-17). This research has been conducted following the ethical requirements established by Greece national board of ethics. The participants played and evaluated voluntarily and anonymously the game. The participants' knowledge of programming varied; some of them had already been well-acquainted with the basics of programming, whereas the majority of them had little or no prior knowledge of programming.

Results

The demographic data of the pilot evaluation are presented in Figure 2.

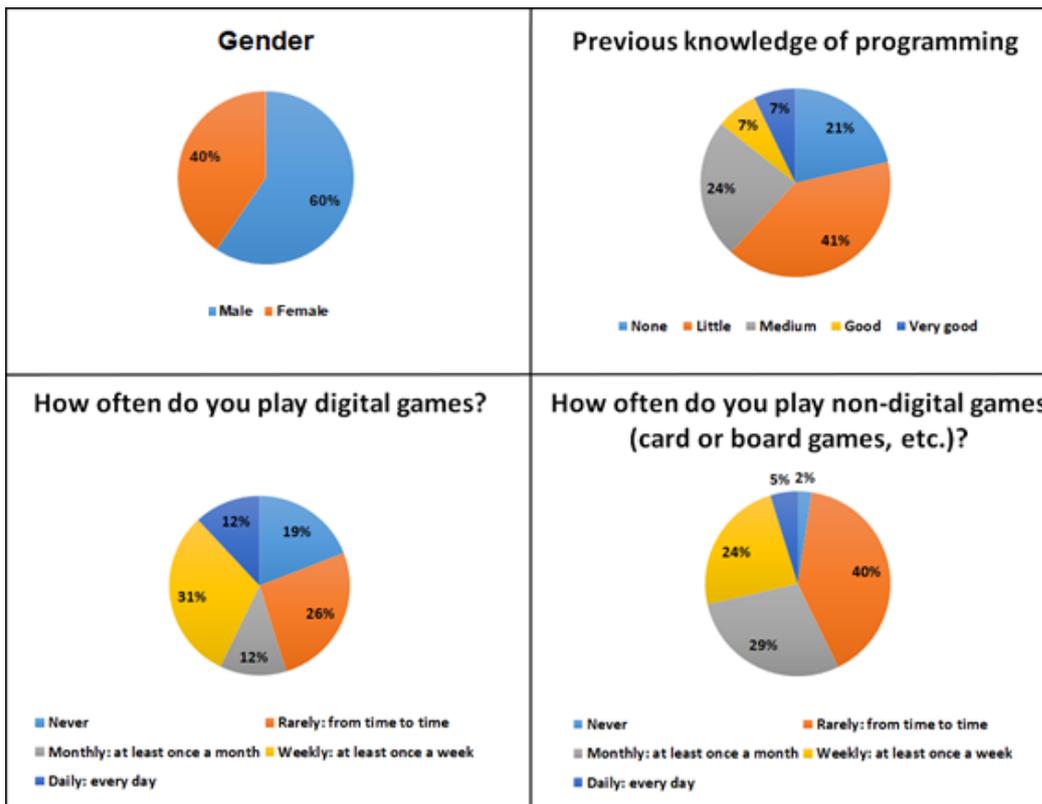


Figure 2. Demographic data of the pilot evaluation.

It is worth noting that the students' level of knowledge in programming varied, while the evaluation also included students that did not have any previous knowledge of programming at all. Specifically, 62% of the students had little or no previous knowledge in programming. This was arranged in order to evaluate the game's ability to teach programming from scratch. Quite surprisingly nearly half the students stated that rarely (26%) or never (19%) play digital games.

In the next paragraph, we analyze the evaluation's results regarding the player experience (usability, confidence, challenge, satisfaction, fun, focused attention, relevance) and perceived learning factors.

Usability. Students' replies regarding the usability of jAVANT-GARDE are presented in Figure 3. The results of the evaluation indicate that the participants found the game design attractive in overall (29% strongly agreed, 52% agreed). The text fonts and colors were considered well blended and consistent. The educational content of the game was rendered more than adequate for its purposes, proving that no previous knowledge, skills nor external support are required in order to advance in the game. Moreover, the participants had no difficulty in playing the game, while they also deemed that the game was easy to play and that most people would learn to play it very quickly. A percentage of 45% of the students strongly agreed and another 45% agreed that the game rules were clear and comprehensible, leading to no misconceptions or difficulty in understanding them. The colors of the game were proved to be meaningful, assisting in the comprehension of the theory as well as the aesthetics of the game. On the other hand, the participants were ambivalent as to whether the game prevented them from making mistakes (median = 0). However, it was clear to them that whenever they made a mistake, it was easy to recover from it quickly (median = 1). We must note that in programming, learning through trial and error is quite a common practice.

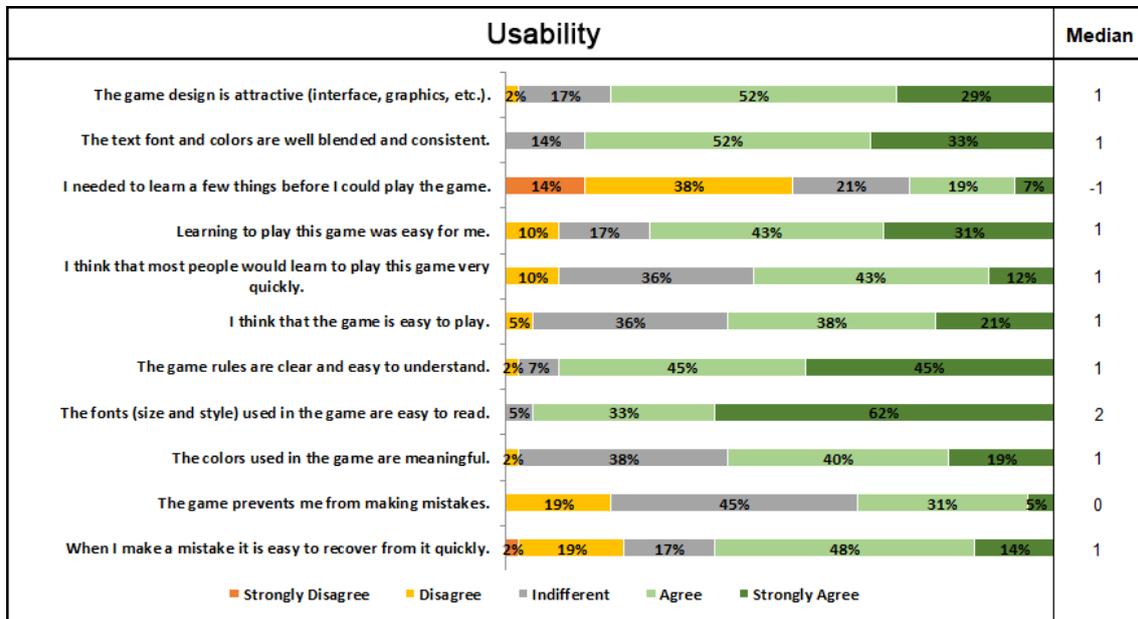


Figure 3. Evaluation results on Player Experience: Usability.

Confidence. The results that refer to the confidence of the students regarding the learning outcomes of the game are presented in Figure 4. At first glance, the participants were not entirely certain whether this game would be easy for them or not (median = 0). Nevertheless, as soon as they started playing, the contents and structure of jAVANT-GARDE inspired confidence in them and assured them that they would learn with it (24% strongly agreed, 50% agreed).

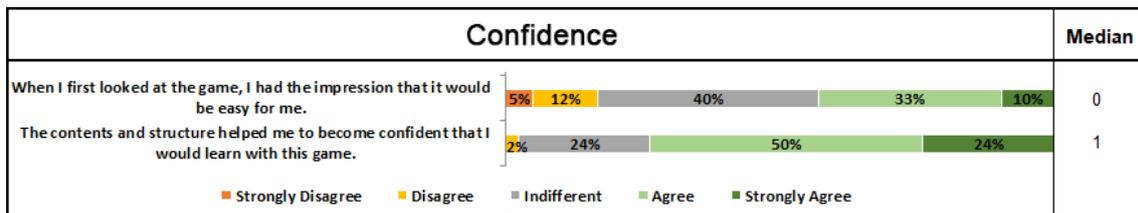


Figure 4. Evaluation results on Player Experience: Confidence.

Challenge. Figure 5 presents the results regarding the challenging aspects of the game. Nearly half the participants (43%) believed that the game was appropriately challenging for them. A lot of effort was put into developing a game that gradually uncovers new and interesting challenges at an appropriate pace and, as it seems, that goal was achieved (14% strongly agreed, 60% agreed). As a result, the game has proven to be engaging for the players and it doesn't cease to captivate their interest as it progresses (median = 1).

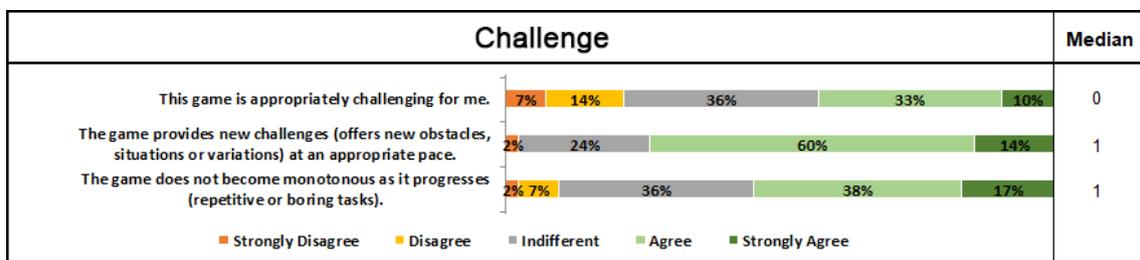


Figure 5. Evaluation results on Player Experience: Challenge.

Satisfaction. The results regarding students' satisfaction during game play are summarized in Figure 6. In terms of satisfaction, the game was evaluated positively. Completing the game's tasks and challenges gave to the players a satisfying feeling of accomplishment (median = 1), boosting their self-esteem. Most of the players felt satisfied with the things that they learned from the game (64%) and they would recommend it to others (55%).

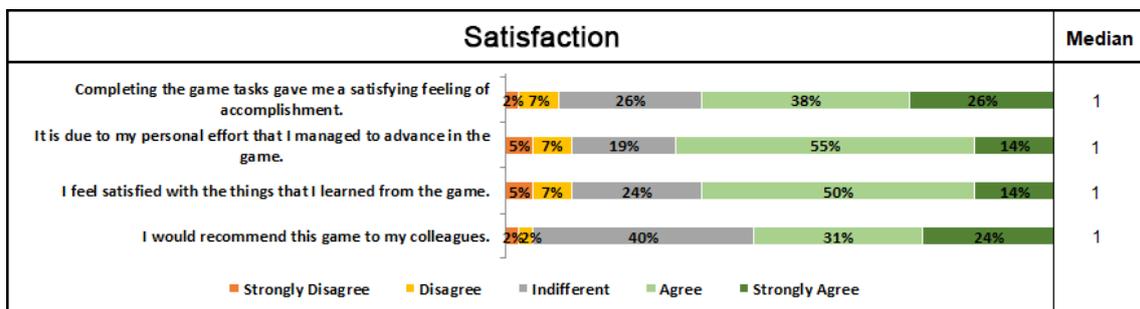


Figure 6. Evaluation results on Player Experience: Satisfaction.

Fun. The results describing the fun that students had while playing jAVANT-GARDE are illustrated in Figure 7. The students enjoyed themselves while playing the game and that can be confirmed from the evaluation as well (31% strongly agreed, 36% agreed). There have been elements in the game that made many players smile, which is a very positive and encouraging sign.

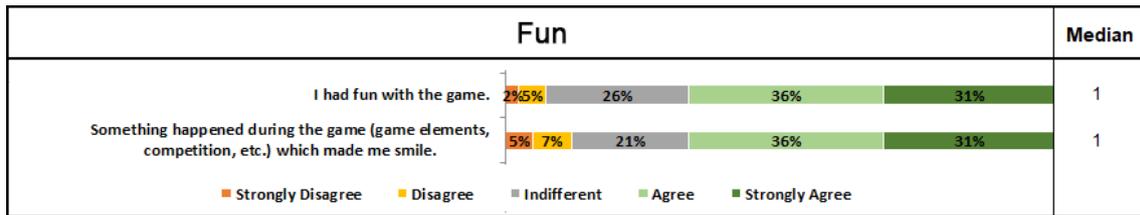


Figure 7. Evaluation results on Player Experience: Fun.

Focused Attention. Figure 8 demonstrates students' answers on how attentive they were, while playing jAVANT-GARDE. The game managed to capture the participants' attention right from the beginning. The results imply that the students were not entirely absorbed while playing the game (median = 0), but this could be partly attributed to the fact that the students were playing all together, at the same time, and right next to each other, making it difficult for them to forget about their immediate surroundings (median = -1).

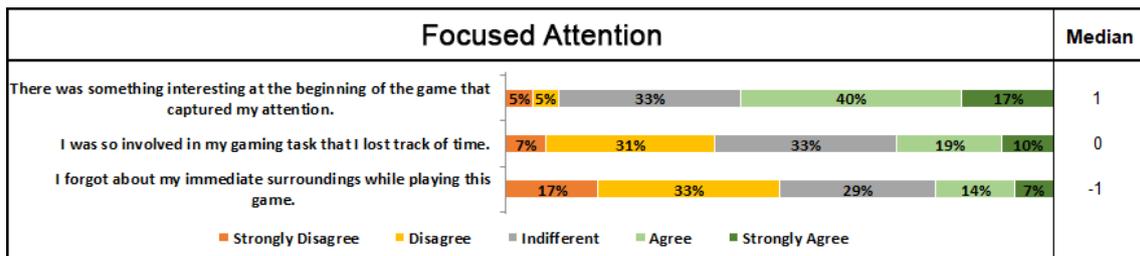


Figure 8. Evaluation results on Player Experience: Focused Attention.

Relevance. Figure 9 exhibits the students' view of how relevant the game is to programming. The students that participated in the evaluation were selected randomly, while some of them never even had a

programming course before. Despite the fact that programming was not relevant to some students' interests (median = 0), they evaluated the game positively in overall, and it can be deduced that they managed to play the game and learn some of the basic concepts of programming. Additionally, they acknowledged the game as an adequate teaching method (33% strongly agreed, 50% agreed) and the majority of them stated that they prefer learning with this game to learning through other ways (43% strongly agreed, 31% agreed).

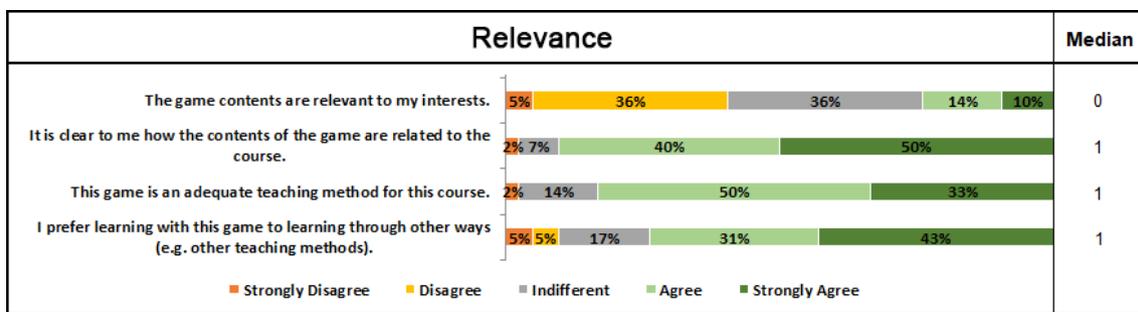


Figure 9. Evaluation results on Player Experience: Relevance.

Perceived Learning. The results that concern the educational purposes in terms of perceived learning are presented in Figure 10. The game contributed to the players' learning on the basic concepts of programming (26% strongly agreed, 43% agreed) and also, it helped them get a better understanding of what they already knew (17% strongly agreed, 50% agreed). Additionally, 38% of the students strongly agreed while 40% agreed that the game allowed for efficient learning compared with other activities regarding the basic concepts of programming, confirming that the game achieved the main goal of educational games.

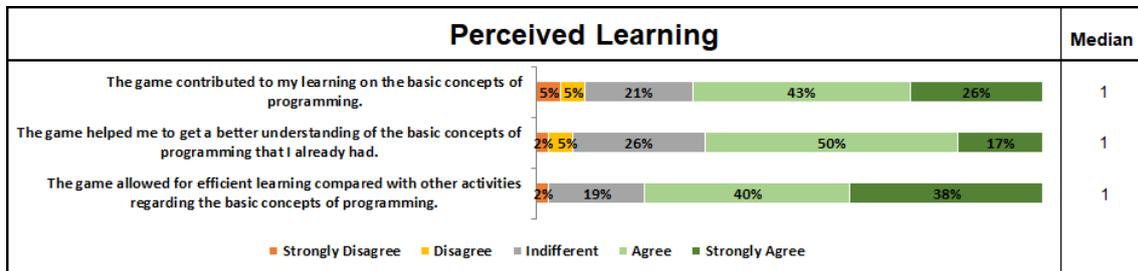


Figure 10. Evaluation results on Perceived Learning.

The feedback from the participants on the comments section was overwhelmingly positive and complimentary with only a few advising for more health points and also, easier handling of the robot – less sliding over the platforms due to momentum. There were conflicting views regarding the graphics of the game as some of the participants considered them as one of the strong aspects of the game, while others stated that they would prefer a more realistic and less playful approach. The former view was dominant nonetheless.

Conclusions

jAVANT-GARDE is an open-source, cross-platform serious game for introducing novices to programming using Java. The pilot evaluation of the game by 42 high-school students gave positive results both in terms of player experience and short-term learning. Regarding the player experience positive results were recorded in terms of usability, confidence in playing the game, challenge, satisfaction, fun, and relevance, while less positive results were recorded in terms of focused attention. What is considered important is that the vast majority of students stated that the game helped in learning new programming concepts and better comprehending concepts they had already been taught. We consider these results rather positive taking into account that the majority of students had little or no programming experience, nearly half rarely or never played entertainment games and several students were not interested in programming.

The results of the first pilot evaluation of jAVANT-GARDE have to be confirmed by a long-term evaluation of the game either in the classroom, or as a self-studying tool. Since the game is open-source anyone interested can make improvements based on the evaluation results, or even extend the game in order to support more programming concepts. Moreover, the game has been designed in such a way that it is possible to add another language for the interaction with the game and provide the educational material in another language (besides English and Greek currently supported) without much effort.

Another interesting possibility is to evaluate the mobile version of jAVANT-GARDE as well and investigate whether mobile learning provides even more motivation to students and generally whether it has a positive effect on learning or not.

Acknowledgements

We would like to thank the 2nd Experimental High School of Thessaloniki; the principal who allowed the conduction of the pilot evaluation, the IT teacher for his cooperation, and all the students that participated in the pilot evaluation.

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

- Clapper, T. (2018). Capitalizing on the Most Important Part of a Learning Session: The Experience. *Simulation & Gaming*, 49(1), 3 – 7. doi: [10.1177/1046878118755155](https://doi.org/10.1177/1046878118755155)
- Crookall, D. (2014). Engaging (in) gameplay and (in) debriefing. *Simulation & Gaming*, 45(4-5), 416-427. doi:[10.1177/1046878114559879](https://doi.org/10.1177/1046878114559879)
- de Freitas, S., & Jarvis, S. (2006). A Framework for Developing Serious Games to meet Learner Needs. Interservice/Industry Training. *Simulation and Education Conference*, Orlando, Florida.
- de Freitas, S. (2006). Learning in immersive worlds: A review of game-based learning. Bristol, UK: JISC e-Learning Programme. Retrieved from http://researchrepository.murdoch.edu.au/id/eprint/35774/1/gamingreport_v3.pdf
- Ibrahim, R., & Jaafar, A. (2009). Educational games (EG) design framework: combination of game design, pedagogy and content modeling. *In the International Conference on Electrical Engineering and Informatics*, (pp. 293-298), 2009, Selangor, Malaysia. doi: [10.1109/ICEEI.2009.5254771](https://doi.org/10.1109/ICEEI.2009.5254771)
- Malliarakis, C., Satratzemi, M., & Xinogalos, S. (2014). Designing educational games for computer programming: A holistic framework. *Electronic Journal of e-Learning*, 12, 281-298.
- 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 Retrieved from: http://www.incod.ufsc.br/wp-content/uploads/2016/07/Relatorio-Tecnico-INCoD_GQS_03_2016_Ev11.pdf
- Salen, K., & Zimmerman, E. (2004). *Rules of play: Game design fundamentals*. Massachusetts London, England: The MIT Press Cambridge.
- Yusoff, A., Crowder, R., Gilbert, L., & Wills, G. (2009). A conceptual framework for serious games. In *Ninth IEEE International Conference on Advanced Learning Technologies, 2009. ICALT 2009* (pp. 21-23). IEEE. doi:[10.1109/ICALT.2009.19](https://doi.org/10.1109/ICALT.2009.19)

Xinogalos, S. (2015). Object Oriented Design and Programming: an Investigation of Novices' Conceptions on Objects and Classes. *ACM Transactions on Computing Education*, Vol. 15, Issue 3, Article 13 (September 2015), 21 pages. doi: [10.1145/2700519](https://doi.org/10.1145/2700519)