

Formative evaluation of an adaptive game for engaging learners of programming concepts in K-12

Renny S. N. Lindberg¹, Teemu H. Laine²

¹Vrije Universiteit Brussel, Renny.Lindberg@vub.ac.be

²Luleå University of Technology, teemu@ubilife.net

Abstract

As the global demand for programmers is soaring, several countries have integrated programming into their K-12 curricula. Finding effective ways to engage children in programming education is an important objective. One effective method for this can be presenting learning materials via games, which are known to increase engagement and motivation. Current programming education games often focus on a single genre and offer one-size-fits-all experience to heterogeneous learners. In this study, we presented Minerva, a multi-genre (adventure, action, puzzle) game to engage elementary school students in learning programming concepts. The game content is adapted to play and learning styles of the player to personalize the gameplay. We conducted a formative mixed-method evaluation of Minerva with 32 Korean 6th grade students who played the game and compared their learning outcomes with 32 6th grade students who studied the same concepts using handouts. The results indicated that, in terms of retention, learning was equally effective in both groups. Furthermore, the game was shown to facilitate engagement among the students. These results, together with uncovered issues, will guide Minerva's further development.

Keywords: game, programming, computer science education, learning style, play style, adaptation, K-12;

1. Introduction

Countries around the world have recently awakened to the increasing need for people with programming skills. Europe alone will have up to 756 000 ICT job vacancies by 2020 [1]. Governmental initiatives and non-profit organizations, such as All you need is C<3DE, Code.org and Codeclub, are trying to stem the tide with a common goal of providing inspiration, motivation, materials and tools for programming education from early ages on. Moreover, 15 European countries have integrated programming into their national K-12 curricula [2], and the Republic of Korea is following this trend starting from 2018 [3].

It is generally agreed that most students find programming challenging and they struggle even with basic concepts [4], [5]. Because of this, it is important to approach the topic of programming education for younger children with care. One way to do this is by introducing the core programming concepts through a game because games can be motivating and engaging educational tools, and they have been shown to give positive results in learning [6]–[8].

The oldest programming education games date back to the 1960s [9]. Vahldick et al. [10] published a survey of programming education games in 2014, and more games have



emerged since. To the best of our knowledge, previous programming education games provided no adaptation or minimal adaptation, and many of them focus on a single genre, such as puzzle. One-size-fits-all is a typical approach in educational game development, thus it can be hard for some players to approach if the type of gameplay does not suit them [11], [12]. Additionally, providing learning content adaptively could promote engagement and improve learning experiences [13].

In this paper, we present the concept and formative evaluation of Minerva, a game that aims to meet the K-12 programming education needs through adaptation of gameplay and learning contents. The adaptation is based on play and learning styles as described by Bartle [14] and Honey and Mumford [15], respectively. After presenting the game concept, we show the results of a formative mixed-method evaluation with Korean 6th grade elementary school students using a questionnaire, interviews and a retention test between the game group (N=32) and the control group (N=32). The purpose of the evaluation is to see how the game engages the learners and how it compares to a traditional learning method in retention, and to identify any issues that may hinder engagement.

2. Background

2.1. Programming Education in K-12

Learning programming can be challenging. McCracken et al. [16] showed that many computer science students could not program after their first year. Bennedsen and Caspersen [17] conducted an international survey on introductory programming course pass rates in 61 institutes, showing that 33% of the enrolled students failed. In a follow-up study, Watson and Li [18], with a doubled sample size, showed similar results with a pass rate mean of 67%. Both studies emphasized that they did not consider the failure rates to be alarmingly high. Watson and Li also remarked that despite years of research there still are no validated reasons to explain the failure rates.

In search for these reasons, Beaubouef and Mason [19] discovered the following causes for high attrition rates among computer science students: 1) poor advising before and after college, 2) poor math skills, 3) poorly designed courses, 4) lack of practice and feedback, 5) using unskilled teacher's assistants, 6) poor management, and 7) bad choices of programming language. Moreover, Jenkins [5] discovered that aptitude, learning styles and motivation are among the factors that could affect the effectiveness of learning to program. These studies show that there is no single reason explaining why students perform poorly or lose interest when studying programming.

In the case of children, other aspects must also be considered. Resnick et al. [20] listed factors that may damage children's enthusiasm to learn programming: 1) too complex programming languages used, 2) programming is introduced with activities that do not interest children, and 3) programming is introduced in a context where guidance is difficult when things go wrong. These factors can be mitigated by teaching the basic concepts without a real programming language, using games to increase interest and engagement, and providing personalized content and guidance via adaptation, respectively.

The interest toward teaching programming at K-12 has been re-invigorated by easy-to-approach visual programming tools, such as Scratch [20]. Lye and Koh [21] reviewed 27 empirical studies on programming in K-12 and higher education. They discovered that K-12 students were typically given access to visual programming tools to create digital stories and that most studies reported positive outcomes.

2.2. Engagement in Educational Games

Student engagement is essential for effective education, as it has shown to counter important academic problems, including low performance, boredom, alienation and drop-outs [22]. Engagement is also one of the core benefits of using games for education [23],

[24], and there exists a link between engagement and motivation [23]. Both engagement and motivation are often mentioned as positive results of using educational games [24], [25], [7], which is why educational game designers must pay attention to developing engaging features in their games [24].

Previous studies have proposed engagement taxonomies, which can be useful when analyzing learner engagement in educational games. Fredricks [26] divided engagement into three types:

Behavioral: Attendance, participation in academic activities, and positive behaviour.

Emotional: Focuses on the positive (and negative) reactions to school, teachers, and activities. Excitement, happiness and interest show positive emotional engagement.

Cognitive: how invested the learner is in learning; the level of effort the learner is willing to exert towards a learning goal.

These three types describe student engagement in school environments. Moreover, Bangert-Drowns and Pyke [27] discussed student engagement with educational software and proposed seven engagement levels (from highest to lowest):

- **Literate thinking:** Learner interprets the content from multiple perspectives and manipulates the software to explore
- **Critical engagement:** Learner manipulates the software to find its limitations and to test personal understanding
- **Self-regulated interest:** Learner creates goals within the software to make it interesting and adapts the software to meet the goals
- **Structure-dependent engagement:** learner is competent with the software, following goals set by the software.
- **Frustrated engagement:** Learner has clear goals but is unable to achieve them due to difficulty with operating the software, thus expressing frustration.
- **Unsystematic engagement:** Learner has unclear goals and moves between activities without necessarily completing them.
- **Disengagement:** Learner actively avoids using the tool.

2.3. Programming Education Game Genres

Videogames can be categorized into genres (e.g. action, adventure, role-playing, simulation and sports), and combining genres is not unusual. When reviewing previous programming education games, we categorized them into six genres: action, adventure, puzzle, role-playing, simulation and strategy. Table 1 exemplifies programming education games, their genres, and how they teach programming. Most games approach programming pedagogy in puzzle format, especially games like Lightbot are common. We intend to publish another article with a thorough review of programming education games.

Table 1. Examples of programming education games

Game	Genres	Programming pedagogy	Source
Code Spells (2013)	Action	Write code to craft spells in a sandbox world.	www.multidimensionalgames.com
Hack'n' Slash (2014)	Action, Puzzle	Explore dungeons by hacking various objects.	www.hacknslashthegame.com
Else Heart.Break (2015)	Adventure	Story-based adventure with hackable game objects.	www.elseheartbreak.com
Code Combat (2013)	Role-playing, Adventure	Build a character and write code to traverse a maze by solving obstacles.	www.codecombat.com
Screeps (2016)	Strategy	Base building by controlling units with JavaScript	www.screeps.com
Slush Smackdown (2015)	Simulation (sport)	Control a boxer by writing code.	www.slushsmackdown.com
Daisy the Dinosaur (2016)	Puzzle	Solve logic puzzles by choosing a correct sequence of commands that control a dinosaur.	www.daisythedinosaur.com
Kodable (2014)	Puzzle	Solve logic puzzles by choosing a correct sequence of commands that control a Pacman-like character.	www.kodable.com
Lightbot (2008)	Puzzle	Solve logic puzzles by choosing a correct sequence of commands that control a robot.	www.lightbot.com
Machineers (2013)	Puzzle	Fix broken machines using basic programming concepts disguised as machine parts.	www.machineers.com
Move the Turtle (2012)	Puzzle	Move a turtle with a Logo-like graphical language.	www.movetheturtle.com

Although focusing on one or two genres makes the development of educational games simpler, the outcome may not appeal to heterogeneous learners because certain personality traits may prefer certain video game genres [28]. This information is important for educational game designers, as using multiple genres may increase the likelihood of positive reception among a heterogeneous group of learners.

2.4. Adaptive Learning Games

Researchers have proposed ways to adapt learning content or play experience in different ways. For example, Kickmeier-Rust et al. [29] propose a formal, computable method for creating educational games with adaptive storytelling. Their system performs story adaptation through modelling of stories, cognitive domain, and the learner. In another study, Magerko [12] listed game adaptation possibilities, including: a) Storytelling, by providing a multi-path story; b) Player motivation, by changing the game according to player types;

and c) Non-Player Characters (NPCs) that adapt to different player types. Magerko's approach to adaptation is to use pedagogically relevant knowledge: "what strategies the player is using to solve certain problems, player learning style, and so on – games for learning can incorporate modelling approaches for both pedagogical and entertainment goals. A game can be tailored to both better engage as well as educate the player by using both kinds of knowledge" [12].

An important goal for adaptive games is the increased engagement among a broad range of players. The target group of many educational games is heterogeneous, which was also acknowledged by Chen [30], who raised the importance of personal flow zones: "to design an interactive experience for a broader audience, the experience cannot be the same for all players or users. Any such experience must offer many choices, adapting to different users' personal flow zones".

Game adaptation can be divided into three methods: (i) Static: Adaptation is done once at the beginning, based on previously gathered data, (ii) Dynamic: The game constantly monitors the gameplay and adapts according to pre-defined parameters [31], and (iii) Hybrid: a combination of the first two approaches. In our survey, we did not discover programming education games that adapt the learning experience or the gameplay with any of these methods.

2.5. Learning and Play Styles

Based on a previous survey on learning style and play style models [32], we chose Honey and Mumford's Learning Style Questionnaire (LSQ) [15] and Bartle's Player Types [14] as the basis of adaptation in Minerva (see the following sections). The reason for two separate models is to handle the needs of the user as a player and as a learner. By supporting adaptation of the gameplay for different play styles instead of a one-size-fits-all approach, the game could appeal to a wider audience of players. Likewise, by providing personalized learning materials, the learning experience could be enhanced. Moreover, our ultimate goal is to allow the styles to change adaptively during the learning process.

2.5.1. Honey and Mumford's Learning Style Questionnaire

Honey and Mumford [15] define four learner types as:

- Activist: Learns by doing; prefers a practical approach.
- Theorist: Learns by an analytical approach; needs to understand the theory first.
- Pragmatist: Needs to see a connection between the theory and the task at hand; prefers applying theory in real life.
- Reflector: An observer who does not directly interact, but rather collects data from multiple sources.

LSQ's popularity and clear terminology were the main reasons for selecting it for Minerva.

2.5.2. Bartle's Player Types

The Bartle's Player Types (BPT) model [14] defines four player types with the following characteristics.

- Killers: prefer acting upon other game entities by using dominance and aggression.
- Achievers: prefer winning by points, achievements, and collectibles in a competitive manner.
- Explorers: prefer discovery (e.g. hidden locations and bugs) driven by curiosity
- Socializers: prefer communication over gameplay

Despite originally aimed to model behavior of players in multiplayer games, Bartle's model has also been used to identify play styles in single player games. We chose BPT because of its popularity and simple structure.

3. Minerva

Minerva is a game designed for teaching programming to elementary school students. Figure 1 illustrates Minerva's gameplay. To describe the game elements that Minerva utilizes (Table 2), we used the Game Bricks model [33] due to its compactness and simplicity, with our extensions to cover interaction elements. In the following, we briefly explain Minerva's concept, pedagogical content, adaptation framework and technical implementation.

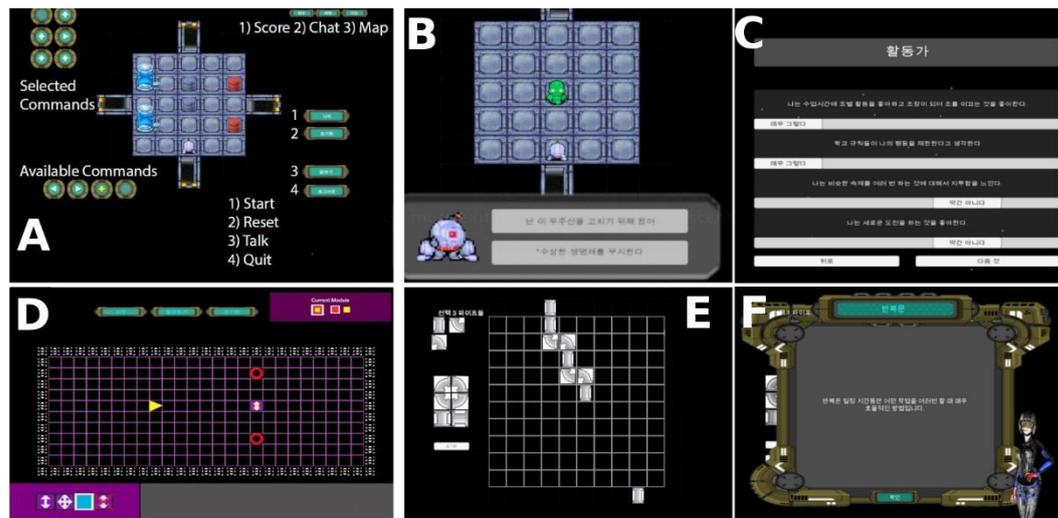


Figure 1. Minerva screenshots: A) Main user interface, B) Dialogue with an alien, C) Questionnaire for play and learning styles, D) Decision puzzle, E) Repetition puzzle, and F) Tutorial view

Table 2. Game elements in Minerva (Game Bricks [37]). Our extensions are under "interaction bricks"

Game Brick	Explanation	Minerva
Avoid	Avoid or evade obstacles or enemies	Navigating the robot past hostile aliens and obstacles
Manage	Collect and maintain resources	Collect points
Shoot	Shooting projectiles at targets	Shooting at aliens and obstacles
Destroy	Destroy game elements	Destructible aliens and obstacles
Match	Match a value with target	Decision and Repetition puzzles (Figure 1D-E)
Write	Write an alphanumeric string as input	Write chat messages
Move	Move physically, or give instructions to move an avatar	Move the robot (Figure 1A)
Select	Select or choose and element	Selecting robot commands (Figure 1A) and NPC dialog options (Figure 1B; Decision and Repetition puzzles (Figure 1D-E)

Interaction bricks		
Interact (player-game)	Interact with game elements, such as NPCs	Have conversations with NPCs
Interact (player-player)	Interact with other players	Chat with other players

3.1. The Game Concept

Minerva is inspired by Lightbot, with some significant modifications. Firstly, instead of having separate puzzles that the player must solve, the game has rooms that are connected by doors and visualized by a game map (Figure 1A). The player may traverse the rooms as they please. A room may have obstacles and aliens to deal with (Figure 1B). Secondly, we tied the game to a story. The player is remotely controlling a robot that has been sent to repair Minerva, a derelict spaceship. This creates a deeper purpose for the player to proceed than simply going from one puzzle to next. The structure of the spaceship also seeps into the design of rooms that represent different parts of the spaceship. Thirdly, the robot and the spaceship rooms (i.e. the main game) teach sequential execution to the player through inputting of command sequences akin to Lightbot. More game mechanics and learning content are provided via puzzles. Each puzzle covers a specific programming concept and they are depicted as “repair missions”, such as fixing Minerva’s fuel lines with a pipe-laying machine. In the current version, there are two puzzles that teach the concepts of repetition (Figure 1E) and decisions (Figure 1D). Fourthly, each room of the spaceship has a chat feature through which players may converse.

Unlike many programming education games that focus on one genre, Minerva mixes genres. It has some characteristics of adventure games, such as story, exploration and interaction with NPCs. Minerva also offers action through destruction of the game environment and its inhabitants. Lastly, the repair missions are puzzles.

3.2. Pedagogical Content

Minerva aims to teach basic programming concepts without coding. We selected five programming concepts – input, output, math, repetition and decisions [34] – to form the game’s pedagogical content. The puzzles cover repetition and decisions (Figure 1D-E), whereas input, output and math are introduced in the story dialog (Figure 1F). All concepts are revised once more at the end. Additionally, the game teaches the player about sequential execution, as the player’s commands to the robot are executed sequentially. This is fundamental for understanding how computer programs execute.

Minerva was built based on the experiential learning theory [35]. However, it can be argued that to some extent Minerva also utilizes constructivism [36], as well as socio-cultural theory [37]. We discuss these in Section 5.

3.3. Adaptation Framework

Minerva adapts its content and gameplay to the player’s learning style and play style (Figure 2). Here we describe the questionnaire for mapping the play and learning styles as well as how the adaption is implemented.

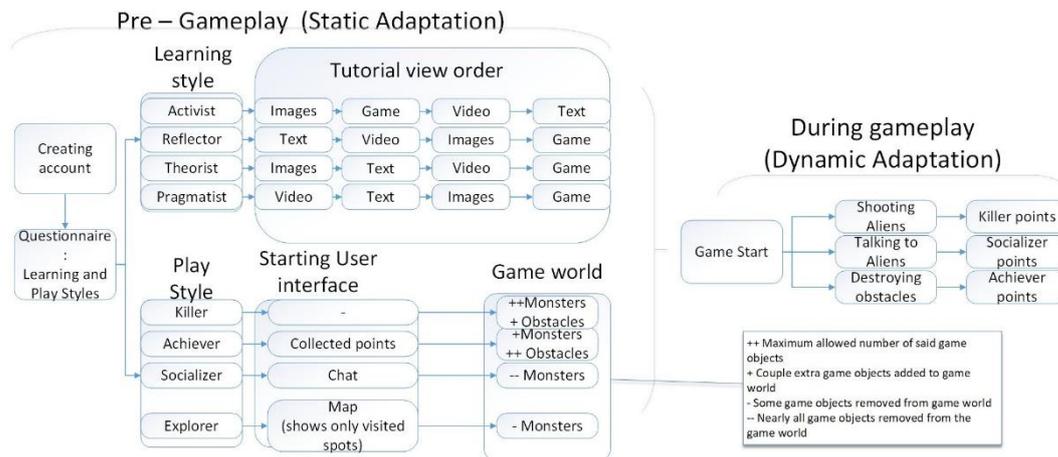


Figure 2. Static and dynamic adaption to play and learning styles

3.3.1. Questionnaire

Based on a previous questionnaire for detecting learning and play styles [32], we created a questionnaire that is filled by the player when they create a new account (Figure 1C). Thus, the questionnaire does not obstruct the gameplay experience. The questionnaire handles the static part of the adaptation, with its results defining the initial play and learning styles for the player.

The questionnaire is divided into learning and play style portions. The questionnaire has four four-point Likert-scale statements for each style, thus 16 questions in total. Dominant learning and play styles are determined by a point system: +2p Strongly agree, +1p Agree, -1p Disagree and -2p Strongly disagree. If the player scores equally in two or more styles within a model, additional statements for each style are shown of which the player can only select one.

3.3.2. Adaptation to Play Style

The play style adaption affects two aspects of the game. Firstly, the numbers of encountered obstacles and alien NPCs may change. For instance, a room may have a varying number of aliens depending on the player's play style: for Killers and Achievers there are three aliens, whereas for Explorers and Socializers there is only one. The second aspect that is adapted is navigational. The game offers the player an in-game chat, scoreboard and map views. Depending on which play style the player has, one of these views is active by default, with exception of Killers who start with no views active. However, the player is free to change the view anytime. Moreover, Explorers see only those map parts that have been visited; other player types see the entire map.

Choosing what game elements to adapt was a design decision based on our simplified representation of Bartle's player types: Killers need something to act upon (shooting monsters, obstacles), Achievers need a way to collect and keep track of points (scoreboard), Socializers must be able to communicate with NPCs and other players (chat, NPC dialogs), and Explorers need a versatile game world with things to discover (fog-of-war in map).

The play style adaptation process follows hybrid adaptation: the player's initial play style is established via the questionnaire, and it can change depending on the player's actions. The game world is sprinkled with interactable game entities, such as aliens that the player can ignore, talk to, or shoot at. Depending on the player's action, points are subtracted/added from/to the corresponding style. For instance, talking adds point towards socializer.

3.3.3. *Adaptation to Learning Style*

Learning style adaptation affects how the learning content is displayed to the player before a puzzle (or another game activity). The learning content is divided into four groups: (i) WHY: Text that offers learning content regarding the topic; (ii) HOW: Video clip that shows how the puzzle is played; (iii) WHAT: Images that show the functions of the puzzle in a greater depth, including also what the player is expected to do; and (iv) DO: the actual gameplay of the puzzle. The order of these groups depends on the player's learning style (Table 3). This adaptation model is based on Magoulas et al.'s [38] work on adaptive navigation support.

Table 3. Learning style adaptation

ACTIVIST	WHAT	DO	HOW	WHY
REFLECTOR	WHY	HOW	WHAT	DO
THEORIST	WHAT	WHY	HOW	DO
PRAGMATIST	HOW	WHY	WHAT	DO

The learning style adaptation process is static. This means that the player's learning style, detected by the questionnaire, remains the same independent of the way the player behaves during the gameplay.

3.4. *Technical Implementation*

Minerva was built using the Unity3D game engine (client) and the Spring Framework (server). The server manages multiple game sessions through queries received from clients. These include registering player details, such as profile and game status, and, once the player logs in, a unique session key that the client uses to authenticate. The game level maps are stored on the server, which are adaptively returned when the client makes a level request. For example, the server decides which obstacles/aliens are to be activated depending on the play style. This is done by creating a unique state of each obstacle and alien for each player in the database.

4. *Formative Evaluation*

We conducted a formative evaluation [39] on the first prototype of Minerva to learn how the design and implementation phases can be improved to increase engagement and retention efficiency. The reader might be interested in a previous study that reported the detected learning and play styles [40].

4.1. *Evaluation Design*

4.1.1. *Participants*

Minerva was evaluated with two sixth grade classes at a Korean elementary school. In the Korean age reckoning system, the participants were 13 years old, i.e. 11-12 years old in western counting. Two classes were randomly assigned to the game group (N: 32, Females: 15, Males: 17) and the control group (N: 32, Females: 17, Males: 15). Only one student (male) in the game group had previous experience about programming.

4.1.2. *Instruments and data collection*

All instruments were first created in English and then translated to Korean. An elementary school teacher validated the language of the translated instruments.

A post-game questionnaire with Likert-scale statements (1= Strongly Disagree, 4 = Strongly agree) and open questions was distributed to the game group. The questionnaire gathered feedback on what the students liked or disliked about the game, what game features were enjoyed, how was their learning experience, what issues they faced, and what their overall opinion about the game was. The questionnaire is based on a previously conducted study of an educational exergame [41].

A semi-structured interview was administered to 12 game group students and their teacher to gather data on what they thought about the game, programming and programming education. Additionally, the teacher was surveyed on her perceptions about the ongoing curriculum reform and programming education games.

A retention test, inspired by Code.org, measured differences in retention of learning content presented inside the game versus in a paper format. We prepared materials for the control group covering the same programming concepts as Minerva. The materials were handed to the control group by their teacher who had been briefed on the materials and could assist the students with any issues with understanding the materials. The retention test was divided into four questions. The first question had the maximum of 8 points. Questions 2 and 3 were binary choices, with a required justification for the chosen answer. The question four had five blanks that the student filled with programming concepts (input, repetition, decision, output and math).

The data was collected over a two-week period (Figure 3). The game play experiment was conducted during a regular computer class on a Thursday. Researchers pre-installed Minerva onto computers, so that the players could create their accounts and play after a brief introduction by the researchers. The gameplay lengths varied between 11-54 minutes, with most the players playing roughly 30 minutes (mean: 35:33 min). The gameplay time excludes the time for completing the play and learning style questionnaire. Due to technical difficulties, six players played only a limited time (9 minutes or less). The questionnaire results of these players were kept, but their gameplay data was discarded. On the following Monday and Tuesday, interviews were conducted, and the educational materials were distributed to the control group on the same Tuesday. The retention test was administered to both groups one week after the respective interventions.

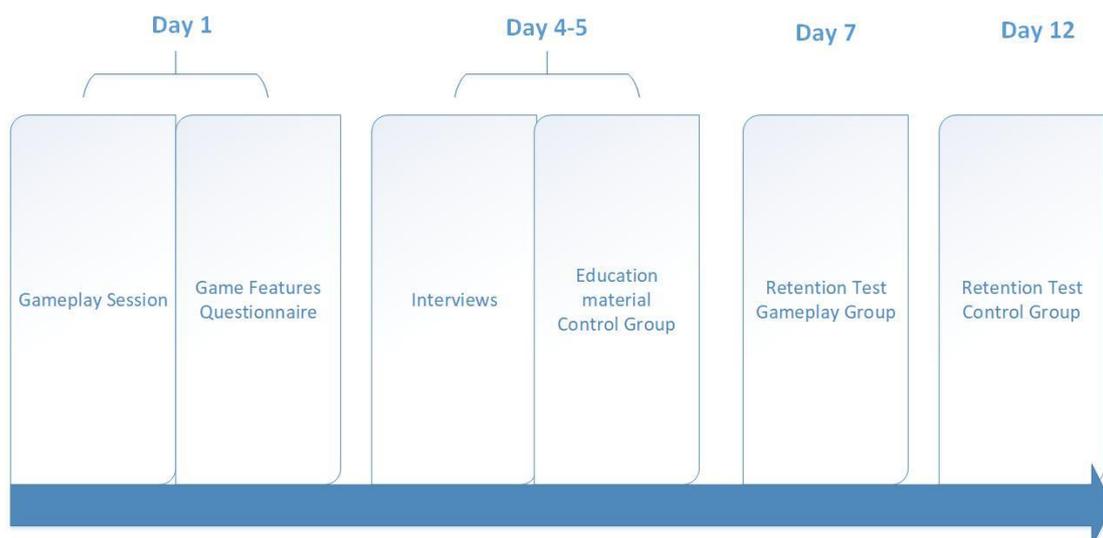


Figure 3. Data collection timeline

4.1.3. Analysis

Statistical analysis of the questionnaire and the retention test data was conducted via calculating, median, mode, mean, standard deviation and effect size. Open questions and interviews were transcribed, translated to English, and analyzed by a content analysis method [42].

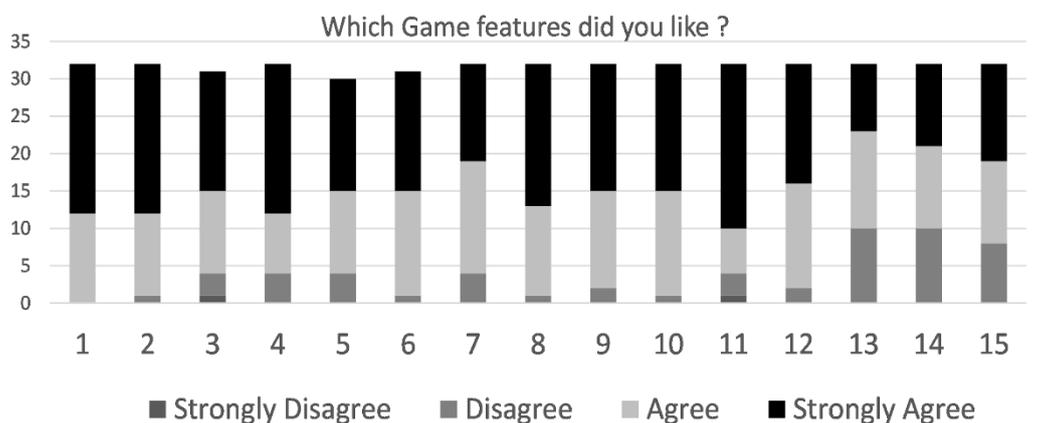
4.2. Results

The results are categorized into: 1) features, 2) learning experience, 3) retention, 4) overall experience, and 5) issues. Quantitative questionnaire data are presented as bar charts with median and mode. Mean (μ) standard deviation (σ) and effect size are reported for the retention test. Qualitative data excerpts are presented to support the results. One game group student's questionnaire and one control group student's retention test was discarded due to incompleteness.

4.2.1. Features

Figure 4 shows the results for the features section of the game group questionnaire, thus indicating the features that were particularly engaging. Many statements received positive responses, except those on presentation of learning contents (statements 12-15), which received nearly 1/3 of disagreements. We also observed that many students were keen to start the actual game, thus skipping over the tutorial views.

- | | |
|---|--------------------------------|
| 1. Issuing commands to the robot | 9. Collecting points |
| 2. Talking to aliens | 10. Exploring the spaceship |
| 3. Shooting aliens | 11. Using the chat |
| 4. Evading obstacles | 12. Reading the game story |
| 5. Solving the decision mission (Data flow) | 13. Reading tutorial texts |
| 6. Solving the repetition mission (Laying fuel pipes) | 14. Watching tutorial pictures |
| 7. Learning useful things | 15. Watching tutorial videos |
| 8. Learning about computer programming | |



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Median	4	4	4	4	3.5	4	3	4	4	4	4	3.5	3	3	3
Mode	4	4	4	4	4	4	3	4	4	4	4	4	3	3	4

Figure 4. Game features. Questions 3, 5 and 6 were left unanswered by a total of three students.

Four students disagreed with statements 4 (evading obstacles), 5 (solving decision mission) and 7 (learning useful things). Some students found moving the robot difficult at the beginning, or boring:

Female student: *"It was hard to figure out how to control the robot."*

Female student: *"The most boring part was going from door to door. The funniest part was sending the data [in the data flow puzzle]"*

Similarly, the decision puzzle, labelled as the "data flow puzzle", was hard for some of the students to understand:

Female student: *"The data flow puzzle needed more explanation, especially how it works and how to fire the data."*

Overall, the students enjoyed the game features. Eight students mentioned that they enjoyed the challenge and the sense of success when they cleared the puzzles, for example:

Female student: *“Good thing is now I know about programming. Making robot's movement was fun and two puzzles were a bit difficult but its difficulty made me to use my brain. It was good.”*

Male student: *“It was great that I could understand the concept of computer working process by playing the pipe and data flow games.”*

Enjoying the sense of success from overcoming challenges can motivate the player, especially when combined with clear and specific goals [43]. Yet Minerva did not fully succeed to provide clear and specific goals:

Male student: *“I got bored first because I couldn't figure out how to play it.”*

Most students enjoyed the additional features built into the game, such the map, the chat and deciding whether to shoot aliens or talk to them. This was evidenced by the questionnaire statements 1-5, as well as open questions and interviews:

Female student: *“I liked that the puzzle games were placed in each side of the last room. I liked that this game has lots of functions that made me have fun, such as check the map, do chatting, talk with aliens etc.”*

Female student: *“1. Graphics were cute, 2. Talking with alien was funny (and they were cute), 3. With some instructions, it was interesting to make robots move by myself, 4. The colorful graphics in the data flow game were pretty.”*

Female student: *“It was fun when I made the robot to move, talk and kill someone.”*

One student selected strongly disagree for statement 3 (shooting aliens). Surprisingly, the same student also answered to an open question as follows:

Male student: *“It would be fun if the aliens would react aggressively even when I select a friendly answer.”*

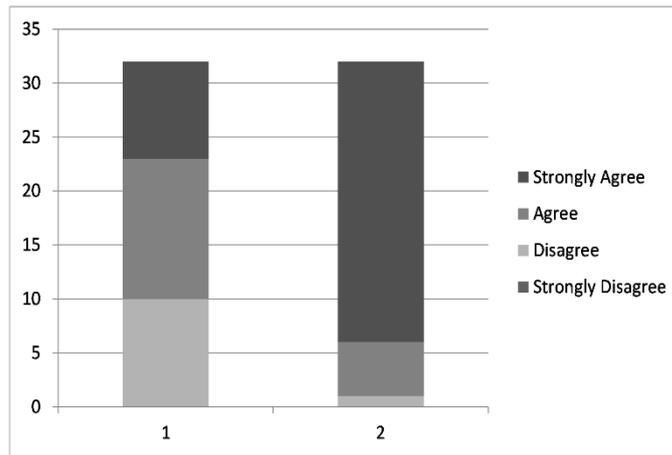
The way the game functions is that if the player talks with an alien and selects non-hostile answers, the alien will leave the room, thus allowing a non-violent conflict resolution. The same student, with two other students, disagreed with the statement 11 (using the chat system).

It can be surmised from these results that most students were engaged in the core features of the game, with some students finding the puzzles and robot control challenging. We also observed that some students were helped by their friends to solve the puzzles.

4.2.2. Game and Learning Experience

Relevance to games and learning statements (Figure 5) queried the students' opinions of the game in comparison to regular digital games and computer classes. Most of the students stated that Minerva was more exciting than regular video games and that playing the game was more exciting than regular computer classes.

- 1 This was more exciting than normal computer or mobile games
- 2 This game was more exciting than a regular computer class

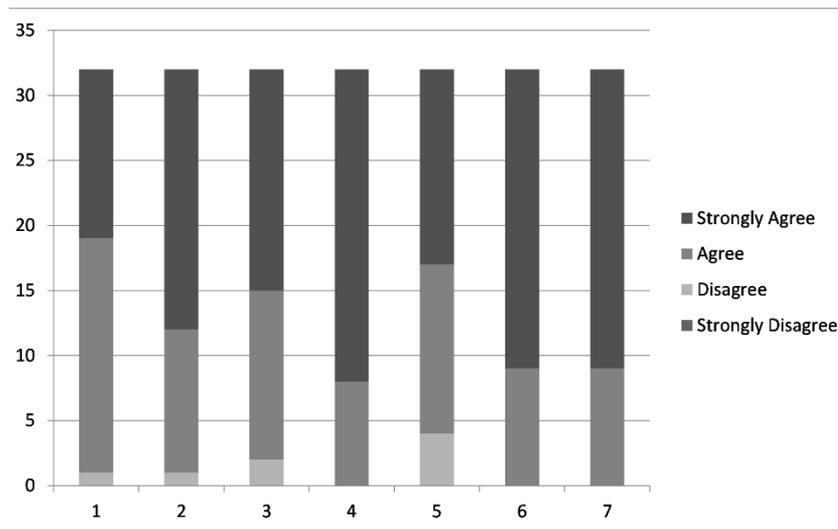


Median	3	4
Mode	3	4

Figure 5. Relevance

We also gathered the students' perceptions regarding their learning experiences with Minerva. Figure 6 shows largely positive responses, thus indicating self-perceived pedagogical potential. Only four questions had a handful of disagreements from the students.

- 1 The game increased my knowledge about computer programming
- 2 The rooms in the game were challenging and stimulating
- 3 The repair missions in the game were helpful for learning
- 4 I gained a sense of achievement through the game
- 5 I will use the knowledge I learned in this game in my life
- 6 The game was worth my effort
- 7 I want to learn by playing this kind of game again



Median	3	4	4	4	3	4	4
Mode	3	4	4	4	4	4	4

Figure 6. Learning experience

4.2.3. Retention

The retention test was divided into four questions in four topics: 1) Issuing commands, 2) Repetition, 3) Decisions, and 4) Basic concepts. After eliminating blank answers, we had 31 and 32 answers from the game group and the control group, respectively. Due to the nature of the questions, only for the first question on “Issuing commands” (maximum 8 points) could the mean, standard deviation and effect size be calculated. P and F stand for “Passed” and “Failed”, respectively.

The first question prompted a set of commands (turn left, turn right, move forward) to move a character from A to B. The student was first given a solution to a similar problem. Both groups did relatively well (Figure 7, Table 4), with three students making mistakes in the game group and eight in the control group.

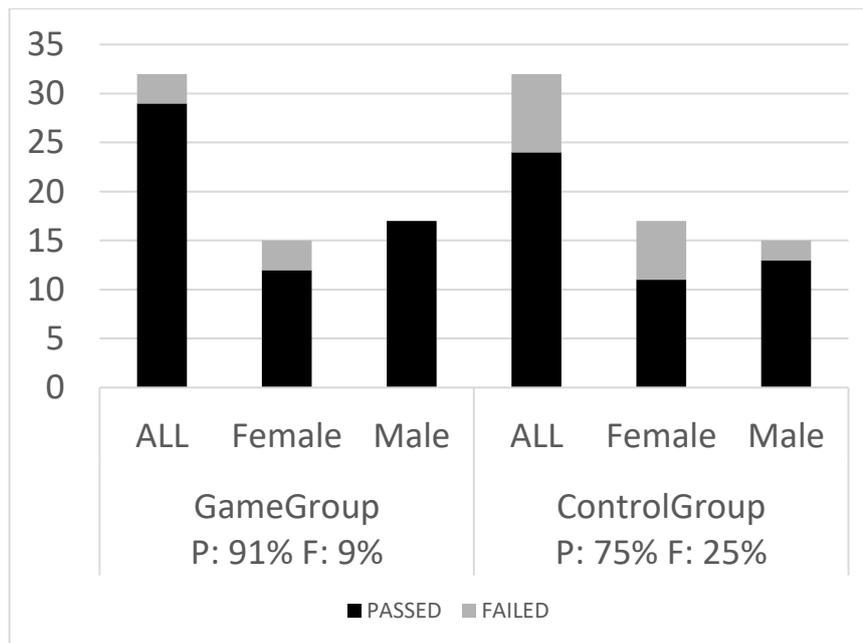


Figure 7. Retention: Issuing commands.

Table 4. Mean, standard deviation and effect sizes on the Issuing commands question

	Game Group			Control Group		
	All	Female	Male	All	Female	Male
μ	7.53	6.69	8.00	7.03	6.53	7.60
σ	1.48	2.31	0.00	1.76	2.09	1.05
Effect size	All	0.28	Female	0.08	Male	0.38

The first part of the second question depicted a character and two code snippets that made the character draw a square. The first code (A) used repetition and the second code (B) did not, and the students were asked to select the more efficient one. In the second part, the students were asked to explain why they selected A or B.

Figure 8 indicates the differences between the two groups, with the game group performing better. Five and twelve students in the game group and in the control group,

respectively, selected the wrong answer (B) or failed to provide a sufficient explanation for selecting A.

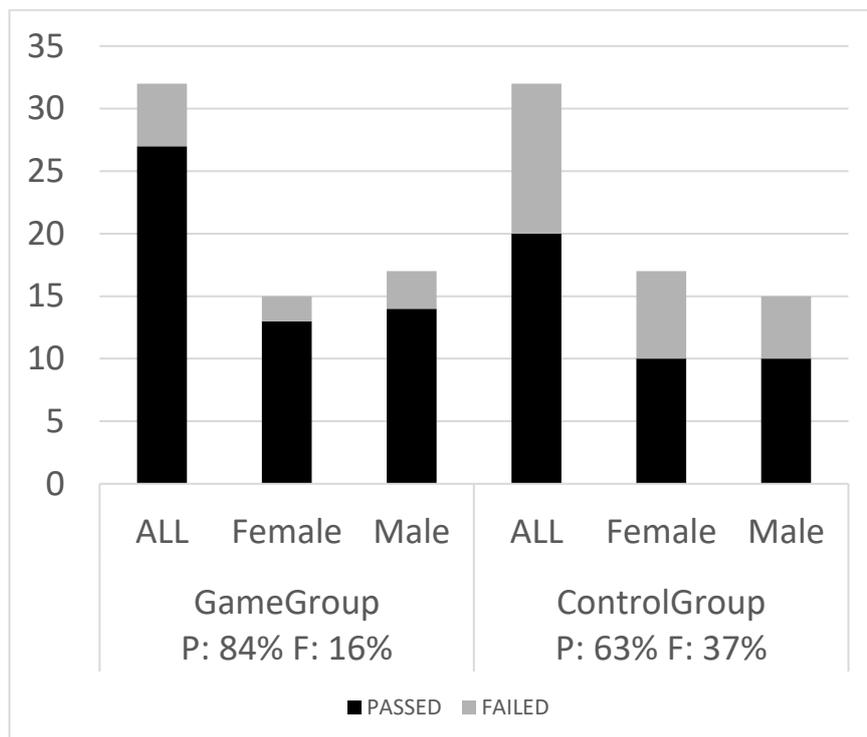


Figure 8. Retention: Repetition

Most students selecting A explained correctly the benefit of repetition, thus understanding how it may be used in programming. For example:

Female student: *“In B, every sentence needs to be inserted one by one, but A doesn’t need those steps because of repetition.”*

Conversely, the students who selected B showed that they either had not understood the code, or did not believe that both solutions do the same thing:

Female student: *“If it is a square, then it needs a lot of steps to make square.”*

Male student: *“B will make a perfect square, but A does not.”*

The third question focused on decisions and was made in a similar manner than the second question (repetition), with two code snippets and an image. Both codes made a bee to move on top of a flower and extract nectar from it. The first code (A) moved the bee on the flower and extracted nectar without any checks. The second code (B) had an IF-check to verify the existence of nectar before extraction.

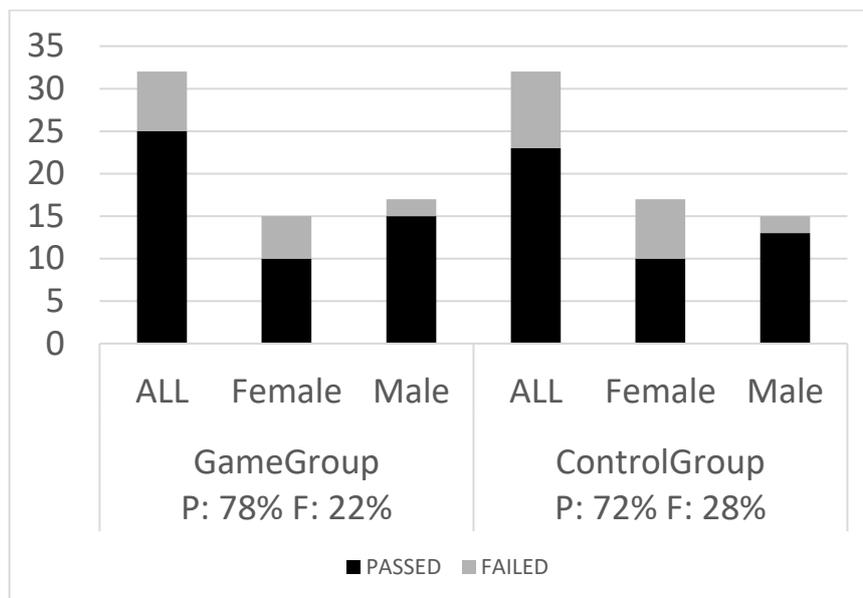


Figure 9. *Retention: Decisions*

As Figure 9 indicates, the groups had similar results, with seven from the game group and nine from the control group selecting A. When justifying their answers, some students preferred A due to its simplicity, and some rejected the logic and math involved:

Female student: *“It can be easily understood because of its simplicity.”*

Female student: *“The flower is already close enough to get, so bee can just try to get honey.”*

Female student: *“It is a simple question. But B will try to solve this problem with a complicated method. It is too mathematical method.”*

Most students who selected B understood why the IF check was beneficial, for example:

Male student: *“It will be a waste of time for the bee when the flower has no nectar, so B is more efficient.”*

The fourth question had a story about the Curiosity rover on Mars using input, output, decision, repetition and math. Parts of the story were blank, and the student was asked to assign the correct concepts to finish it. The results in Figure 10 show that the control group outperformed the game group: 12 game group students made mistakes whereas five control group students failed. Moreover, this is the only retention test question where females outperformed males.

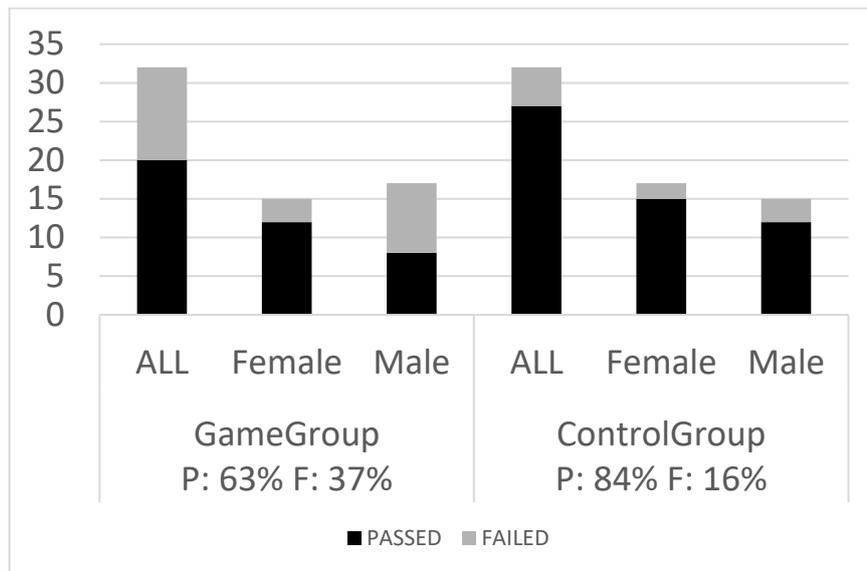


Figure 10 Retention: basic concepts

4.2.4. Overall Experience

We gathered the players' overall perceptions about the game with ternary choices such as: "The Game was... Ugly, Neither, Pretty". According to Figure 11, "neither" was clearly preferred in statements 1-3. For example, for the statement of the game being difficult or easy (1), "neither" was selected by 18 students. In this statement, "neither" suggests that difficulty was just right. Similarly, the game's length (2) was mostly appropriate, with 16 students selecting "neither". The positive results on "Excellent" vs "Bad" (4), "Exciting" vs "Boring" (5) and "Useful" vs "Useless" correlate with the overall mood observed during the experiment, which also the teacher commented upon:

Teacher: "It was surprising in a good way to see those who have normally low motivation for learning were also enjoying the class."

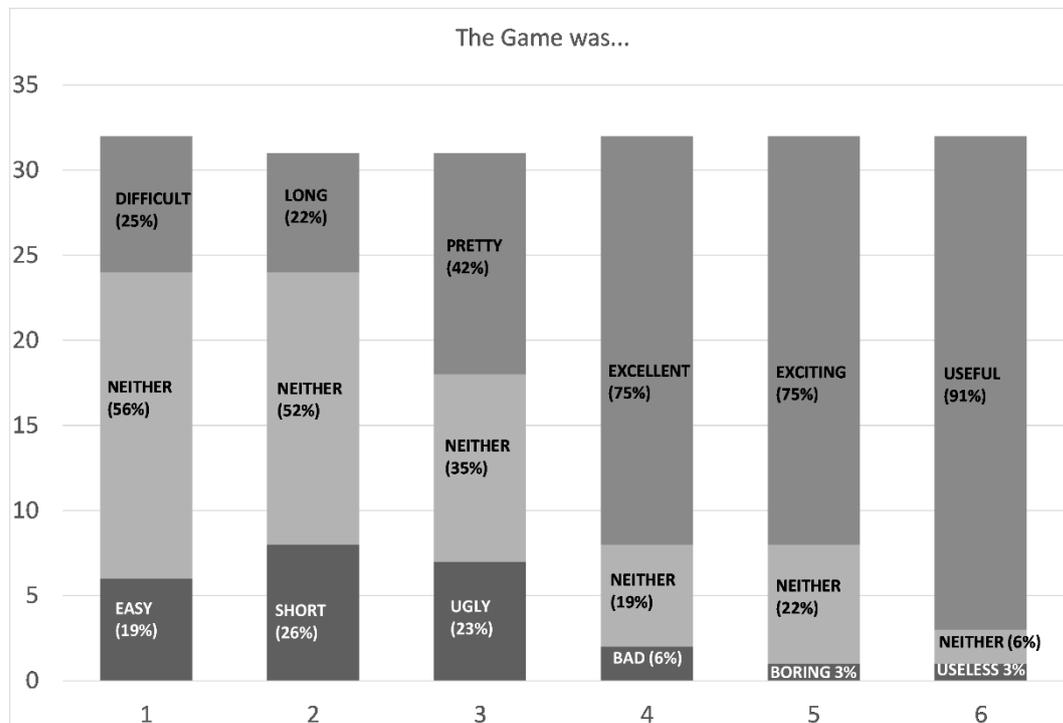


Figure 11. Overall experience. Questions 2 and 3 were left unanswered by one student.

4.2.5. Issues

In the questionnaire, we asked about any issues that the students encountered while playing Minerva (Figure 12). The biggest issue was not knowing what to do next (6) with 16 students supporting this statement. Eight students also agreed and three students strongly agreed that the tutorials were difficult (5), and 9 students stated that instructions were unclear (2).

Moreover, 13 students considered the repetition puzzle to be difficult (7), and seven agreed and one strongly agreed that the decision puzzle was difficult (8). This is aligned with our observations of several students skipping the tutorial views. This student's comment illustrates the puzzle difficulty:

Female student: *"It was little hard to understand the pipe and data flow puzzles."*

During interviews, some students also pointed out learning curve, unappealing tutorials and complex terminology:

Male student: *"It was confusing at first, but when I adapted it was easy to control the robot."*

Female student: *"Reading tutorials was boring but moving and talking to aliens was fun."*

Female student: *"It was hard to understand about the computer terminology."*

Furthermore, 12 students experienced problems with controlling the robot (10). Researchers also observed that some students confused the buttons for turning the robot left or right as commands that would make the robot move left or right. The following excerpts from the students illustrate this:

Female student: *"...It would be better if the tutorial has more explanation about robot control, obstacles, etc."*

Female student: *"The robot did not go to the direction that I ordered."*

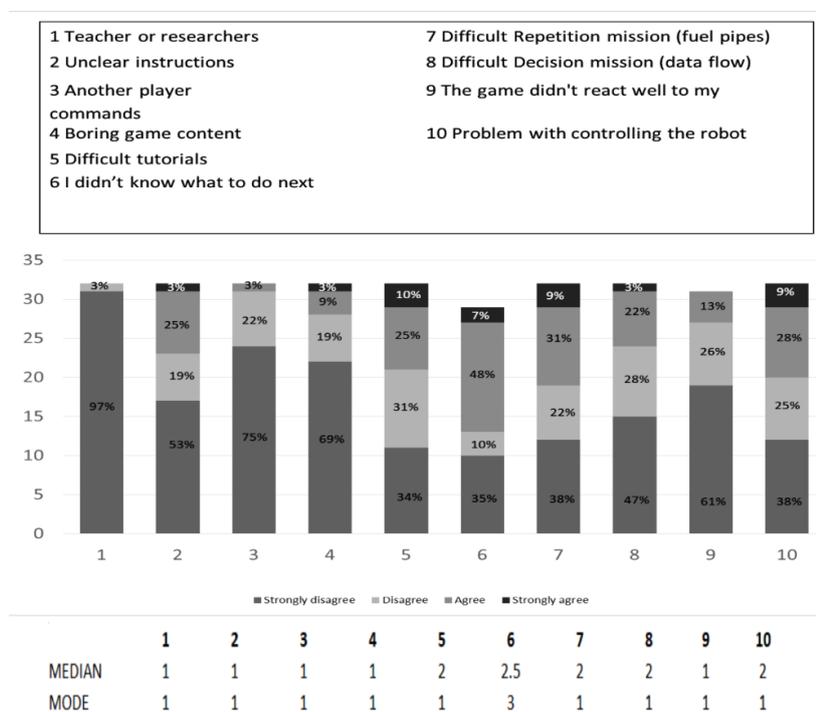


Figure 12. Issues with the game. Questions 6 and 9 were left unanswered by total of four students

Unclear instructions were also noted by the teacher through constructive criticism:

Teacher: “*The explanations of how to play games and how to clear the missions could be enhanced [...] The feature of selecting a difficulty level will be useful in many ways.*”

5. Discussion and Limitations

Programming education for K-12 is here to stay and the number of countries adopting programming education to their curricula will likely increase. Games such as Minerva could alleviate curriculum reforms by making the process of learning programming concepts more engaging. The use of game features from multiple genres and adaptivity aim to increase engagement among heterogeneous learners.

Based on our analysis of the experiment results, we discovered evidence of the players' engagement (and lack of it). In Table 5, we present this evidence through Fredricks' engagement types [26] and Bangert-Drowns and Pyke's taxonomy of engagement [27] (see Section 2.2). Although the game engaged many learners, it could be improved by diminishing frustrated and unsystematic engagements by fixing discovered issues. Conversely, literate thinking and critical engagement should be supported, for example by allowing opportunities for independent exploration and experimentation. Finally, long-term positive emotional engagement, which can help building sustainable use, should be nurtured through the core drives that makes games motivating [44].

Table 5. Engagement in Minerva

Engagement type	Evidence
Fredricks [26]	
Behavioral	Most players were attentive and enjoyed the game features (e.g. talking, killing, moving the robot). Some students also helped others to proceed. According to the teacher, also the students who usually have low motivation, showed enjoyment towards the game.
Emotional	Excitement and interest, which indicate emotional engagement [26], were visible in the data (e.g. game features, comparison to normal class, interview answers). Yet there were a few students who expressed frustration (see below).
Cognitive	Cognitive engagement is observable from the scores of the retention test. The players were mostly keenly focused on solving the puzzles to pass the game. This was also true with the players who initially had problems with controlling the robot.
Bangert-Drowns and Pyke [27]	
Literate thinking	There was no opportunity (too short time) to interpret the game from multiple perspectives.
Critical engagement	Despite short play time, some students had ideas how the gameplay should be changed (talking friendly, then shooting; improve tutorials).
Self-regulated interest	A few players were observed to enjoy the game so much that they played it through more than once. Helping others spontaneously is also a manifestation of self-regulated interest.
Structure-dependent engagement	Most players achieved the tasks in the game without additional help.
Frustrated engagement	A few players expressed frustration when facing technical difficulties or being confused over the game controls.
Unsystematic engagement	All players were aware of the overall goal of the game. The players experiencing problems with controls had unsystematic engagement (e.g. watching others, chatting) until they learned the controls.
Disengagement	We did not encounter any cases of complete disengagement.

The retention test indicated that Minerva reached a similar level of learning than the traditional method, though the former is arguably more enjoyable to the learner. We expect that retention test results will improve in the future after modifying the game based on these results.

As mentioned above, lack of interest in the tutorials was observed. As it stands, we cannot confirm whether learning outcomes would have improved had the tutorials been more interesting. Making the tutorials more engaging is important to tackle in the future development of the game.

Due to the small sample size and the imbalance between detected play and learning styles, we did not analyze the differences between play and learning style groups (e.g. differences regarding liked features). Our previous study [32] suggested that Korean children have tendencies towards certain play and learning styles. The data set must therefore be significantly larger for conducting reliable per-style analysis.

As we mentioned in Section 3.2, Minerva utilizes multiple learning theories. Despite leaning on Kolb's experiential learning theory as the basis of LSQ, Minerva also utilizes patterns seen in constructivism, in which the learner constructs knowledge from experiences and activities. Likewise, some patterns from socio-cultural theory are implemented with the chat. Moreover, several students helped their classmates or observed from others how the game works.

The study has several limitations. Firstly, the formative evaluation focused on engagement and retention, thus adaptive features of the game were not measured. Secondly, the evaluation results may not be applicable across cultures because all test participants were Korean. Thirdly, the evaluation was conducted in a single session, thus long-term effects of the game are unknown. Lastly, the differences between style groups were not analyzed. These limitations will be tackled in future research.

6. Conclusion

We presented a novel programming education game, Minerva, that combines features from multiple game genres and adapts the gameplay and learning contents based on play and learning styles, respectively. We also presented the results of a formative mixed-method evaluation of the game at a Korean elementary school. The gameplay results showed positive reception and engagement among the students, and the retention test results were encouraging, with equal performance to a traditional method. These results may be useful for educational game designers to create more engaging games for programming education and other subjects.

As we indicated with Table 1, most programming education games tend to focus on only one or two genres, and we did not find evidence of adaptation in previous games. Minerva's wider utilization of genres makes it potentially engaging to a wider range of students, and the adaptations of gameplay and learning content have the potential to offer effective learning experiences. However, as the evaluation results indicated, additional development and research are required to fulfill these potentials.

Although Minerva is currently aimed at elementary education, the game could be utilized throughout K-12 with different levels of content. Additionally, Minerva's story-driven approach could increase students' awareness of the usefulness of programming, thus increasing their interest in computer science. To facilitate these visions, we seek to develop

a content editor that would allow educators to customize Minerva to their pedagogical needs.

Acknowledgments

We thank Seungchan Jung, Dohee Kang, Woojeong Kim, Wonjun Jin, Jonghyun Park, Aziz Hasanov, Joochan Kim and Yuna Oh for their invaluable help.

References

- [1] Hüsing, T., Korte, W. B., and Dashja, E., "e-Skills in Europe: Trends and Forecasts for the European ICT Professional and Digital Leadership Labour Markets (2015-2020)," Bonn, 2015.
- [2] Balanskat, A. and Engelhardt, K., "Computing our future: Computer programming and coding Priorities, school curricula and initiatives across Europe," European Schoolnet, Brussels, Belgium, 2015.
- [3] Benedis-Grab, G., "Coding in Schools," vol. 20, no. 4, pp. 1075–1083, 2014.
- [4] Gomes, A. and Mendes, A. J., "Learning to program - difficulties and solutions," in International Conference on Engineering Education, 2007, no. January 2007.
- [5] Jenkins, T., "On the difficulty of learning to program," in Annual Conference of the LTSN Centre for Information and Computer Sciences, 2002, pp. 53–58.
- [6] Connolly, T. M., Boyle, E. A., MacArthur, E., Hainey, T., and Boyle, J. M., "A systematic literature review of empirical evidence on computer games and serious games," *Comput. Educ.*, vol. 59, no. 2, pp. 661–686, 2012. <https://doi.org/10.1016/j.compedu.2012.03.004>
- [7] Boyle, E. A., Connolly, T. M., Hainey, T., and Boyle, J. M., "Engagement in digital entertainment games: A systematic review," *Comput. Human Behav.*, vol. 28, no. 3, pp. 771–780, May 2012. <https://doi.org/10.1016/j.chb.2011.11.020>
- [8] Voulgari, I., Komis, V., and Sampson, D. G., "Learning outcomes and processes in massively multiplayer online games: exploring the perceptions of players," *Educ. Technol. Res. Dev.*, vol. 62, no. 2, pp. 245–270, Apr. 2014. <https://doi.org/10.1007/s11423-013-9312-7>
- [9] McIlroy, M. D., and Vyssotsky, V., "Darwin, a game of survival of the fittest among programs," *Software, Pract. Exp.*, vol. 2, no. 1972, pp. 91–96, 1972.
- [10] Vahldick, A., Mendes, A. J., and Marcelino, M. J., "A review of games designed to improve introductory computer programming competencies," in *Frontiers in Education Conference*, 2014, pp. 1–7, <https://doi.org/10.1109/FIE.2014.7044114>.
- [11] Charles, D. *et al.*, "Player-centred game design: Player modelling and adaptive digital games," in *DiGRA 2005 Conference*, 2005, pp. 285–298.
- [12] Magerko, B., "Adaptation in Digital Games," *Computer (Long Beach, Calif.)*, vol. 41, no. 6, pp. 87–89, Jun. 2008. <https://doi.org/10.1109/MC.2008.172>
- [13] Hwang, G.-J., Sung, H.-Y., Hung, C.-M., Huang, I., and Tsai, C.-C., "Development of a personalized educational computer game based on students' learning styles," *Educ. Technol. Res. Dev.*, vol. 60, no. 4, pp. 623–638, 2012. <https://doi.org/10.1007/s11423-012-9241-x>
- [14] Bartle, R., "Hearts, clubs, diamonds, spades: Players who suit MUDs," *J. MUD Res.*, vol. 1, no. 1, 1996.
- [15] Honey, P. and Mumford, A., *The Manual of Learning Styles*, 3th ed. Peter Honey Publications, 1982.
- [16] McCracken, M., *et al.*, "A Multi-national, Multi-institutional Study of Assessment of Programming Skills of First-year CS Students," *SIGCSE Bull.*, vol. 33, no. 4, pp. 125–180, 2001. <https://doi.org/10.1145/572139.572181>
- [17] Bennedsen, J., and Caspersen, M. E., "Failure rates in introductory programming," *ACM SIGCSE Bull.*, vol. 39, no. 2, pp. 32–36, 2007. <https://doi.org/10.1145/1272848.1272879>
- [18] Watson, C., and Li, F., "Failure rates in introductory programming revisited," *Proc. 2014 Conf. Innov. Technol. Comput. Sci. Educ.*, pp. 39–44, 2014.

- <https://doi.org/10.1145/2591708.2591749>
- [19] Beaubouef, T., and Mason, J., "Why the high attrition rate for computer science students," *ACM SIGCSE Bull.*, vol. 37, no. 2, p. 103, 2005. <https://doi.org/10.1145/1083431.1083474>
- [20] Resnick, M. et al., "Scratch," *Commun. ACM*, vol. 52, no. 11, p. 60, Nov. 2009. <https://doi.org/10.1145/1592761.1592779>
- [21] Lye, S. Y., and Koh, J. H. L., "Review on teaching and learning of computational thinking through programming: What is next for K-12?," *Comput. Human Behav.*, vol. 41, pp. 51–61, Dec. 2014. <https://doi.org/10.1016/j.chb.2014.09.012>
- [22] Fredricks, J. A., Blumenfeld, P. C., and Paris, A. H., "School engagement: potential of the concept: state of the evidence," *Rev. Educ. Res.*, vol. 74, no. 1, pp. 59–109, 2004. <https://doi.org/10.3102/00346543074001059>
- [23] Sedano, C. I., Leendertz, V., Vinni, M., Sutinen, E., and Ellis, S., "Hypercontextualized Learning Games: Fantasy, Motivation, and Engagement in Reality," *Simul. Gaming*, vol. 44, no. 6, pp. 821–845, 2013. <https://doi.org/10.1177/1046878113514807>
- [24] Hamari, J., Shernoff, D. J., Rowe, E., Coller, B., Asbell-Clarke, J., and Edwards, T., "Challenging games help students learn: An empirical study on engagement, flow and immersion in game-based learning," *Comput. Human Behav.*, vol. 54, no. January 2016, pp. 170–179, Jan. 2016.
- [25] Garris, R., Ahlers, R., and Driskell, J. E., "Games, Motivation, and Learning: A Research and Practice Model," *Simul. Gaming*, vol. 33, no. 4, pp. 441–467, 2002. <https://doi.org/10.1177/1046878102238607>
- [26] Fredricks, J. A., "Engagement in School and Out-of-School Contexts: A Multidimensional View of Engagement," *Theory Pract.*, vol. 50, no. 4, pp. 327–335, Oct. 2011. <https://doi.org/10.1080/00405841.2011.607401>
- [27] Bangert-Drowns, R. L. and Pyke, C., "A taxonomy of student engagement with educational software: An exploration of literate thinking with electronic text," *J. Educ. Comput. Res.*, vol. 24, no. 3, pp. 213–234, 2001. <https://doi.org/10.2190/0CKM-FKTR-0CPF-JLGR>
- [28] Zammito, V. L., "Gamers' Personality and their Gaming Preferences," Simon Fraser University, 2010.
- [29] Kickmeier-Rust, M. D., Augustin, T., and Albert, D., "Personalized storytelling for educational computer games," *Lect. Notes Comput. Sci.*, vol. 6944, pp. 13–22, 2011. https://doi.org/10.1007/978-3-642-23834-5_2
- [30] Chen, J., "Flow in games (and everything else)," *Commun. ACM*, vol. 50, no. 4, p. 31, Apr. 2007. <https://doi.org/10.1145/1232743.1232769>
- [31] Charles, D. and Black, M., "Dynamic Player Modelling: A Framework for Player-centred Digital Games," *Proc. 5th Int. Conf. Comput. Games Artif. Intell. Des. Educ.*, vol. Microsoft, no. April, pp. 29–35, 2004.
- [32] Lindberg, R. S. N. and Laine, T. H., "Detecting Play and Learning Styles for Adaptive Educational Games," in *International Conference on Computer Supported Education*, 2016, vol. 1, pp. 181–189. <https://doi.org/10.5220/0005858501810189>
- [33] Djaouti, D., Alvarez, J., Jessel, J.-P., Methel, G., and Molinier, P., "A Gameplay Definition through Videogame Classification," *Int. J. Comput. Games Technol.*, vol. 2008, pp. 1–7, 2008. <https://doi.org/10.1155/2008/470350>
- [34] Downey, A., *Think Python: How to Think Like a Computer Scientist*, version 2.0.17., vol. 2. Needham, MA: Green Tea Press, 2012.
- [35] Kolb, D. A., *Experiential learning: experience as the source of learning and development*. Englewood Cliffs, NJ: Prentice Hall, 1984.
- [36] Ackermann, E., "Piaget's Constructivism, Papert's Constructionism: What's the difference?," in *Constructivism: Uses and Perspectives in Education*, 2001, pp. 1–11.
- [37] John-Steiner, V. and Mahn, H., "Sociocultural approaches to learning and development: A Vygotskian framework," *Educ. Psychol.*, vol. 31, no. 3–4, pp. 191–206, 1996. <https://doi.org/10.1080/00461520.1996.9653266>
- [38] Magoulas G., Papanikolaou, K., and Grigoriadou, M., "Adaptive web-based learning: accommodating individual differences through system's adaptation," *Br. J. Educ. Technol.*, vol. 34, no. 4, pp. 511–527, 2003. <https://doi.org/10.1111/1467-8535.00347>
- [39] Flagg, B. N., *Formative evaluation for educational technologies*. New York: Routledge, 1990.
- [40] Lindberg, R. S. N., Hasanov, A., and Laine, T. H., "Improving play and learning style adaptation in a programming education game," in *International Conference on Computer*

- Supported Education, 2017, vol. 1, pp. 450–457.
<https://doi.org/10.5220/0006350304500457>
- [41] Lindberg, R., Seo, J., and Laine, T. H., "Enhancing Physical Education with Exergames and Wearable Technology," *IEEE Trans. Learn. Technol.*, vol. 9, no. 4, pp. 328–341, 2016.
<https://doi.org/10.1109/TLT.2016.2556671>
- [42] Hsieh, H.-F. and Shannon, S. E., "Three Approaches to Qualitative Content Analysis," *Qual. Health Res.*, vol. 15, no. 9, pp. 1277–1288, Nov. 2005.
<https://doi.org/10.1177/1049732305276687>
- [43] Locke, E. A. and Latham, G. P., *A theory of goal setting & task performance*. Englewood Cliffs, NJ, US: Prentice-Hall, Inc, 1990.
- [44] Chou, Y. K., *Actionable Gamification: Beyond Points, Badges, and Leaderboards*. Octalysis Media, 2015.