computers

MDPI

Article

Learning Analytics to Guide Serious Game Development: A Case Study Using Articoding

Antonio Calvo-Morata *, Cristina Alonso-Fernández, Julio Santilario-Berthilier, Iván Martínez-Ortiz and Baltasar Fernández-Manjón *

Department of Software Engineering and Artificial Intelligence, Complutense University of Madrid, 28040 Madrid, Spain

* Correspondence: acmorata@ucm.es (A.C.-M.); balta@fdi.ucm.es (B.F.-M.)

Abstract: Serious games are powerful interactive environments that provide more authentic experiences for learning or training different skills. However, developing effective serious games is complex, and a more systematic approach is needed to create better evidence-based games. Learning analytics—based on the analysis of collected in-game user interactions—can support game development and the players' learning process, providing assessment information to teachers, students, and other stakeholders. However, empirical studies applying and demonstrating the use of learning analytics in the context of serious games in real environments remain scarce. In this paper, we study the application of learning analytics throughout the whole lifecycle of a serious game, in order to assess the game's design and players' learning using a serious game that introduces basic programming concepts through a visual programming language. The game was played by $N = 134$ high school students in two 50-min sessions. During the game sessions, all player interactions were collected, including the time spent solving levels, their programming solutions, and the number of replays. We analyzed these interaction traces to gain insights that can facilitate teachers' use of serious games in their lessons and assessments, as well as guide developers in making possible improvements to the game. Among these insights, knowing which tasks students struggle with is critical for both teachers and game developers, and can also reveal game design issues. Among the results obtained through analysis of the interaction data, we found differences between boys and girls when playing. Girls play in a more reflexive way and, in terms of acceptance of the game, a higher percentage of girls had neutral opinions. We also found the most repeated errors, the level each player reached, and how long it took them to reach those levels. These data will help to make further improvements to the game's design, resulting in a more effective educational tool in the future. The process and results of this study can guide other researchers when applying learning analytics to evaluate and improve the educational design of serious games, as well as supporting teachers—both during and after the game activity—in applying an evidence-based assessment of the players based on the collected learning analytics.

Keywords: computational thinking; learning analytics; programming learning; serious games; visual programming

check for updates

Academic Editor: Ananda Maiti

Received: 14 February 2025
Revised: 18 March 2025
Accepted: 20 March 2025
Published: 27 March 2025

Citation: Calvo-Morata, A.; Alonso-Fernández, C.; Santilario-Berthilier, J.; Martínez-Ortiz, I.; Fernández-Manjón, B. Learning Analytics to Guide Serious Game Development: A Case Study Using Articoding. *Computers* **2025**, *14*, 122. https://doi.org/10.3390/computers14040122

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Serious games are games designed for a main purpose other than pure entertainment, such as learning, raising awareness, or changing players' behaviors [1]. The efficacy of serious games has been proven in multiple fields, including medicine, science, economics, and literature, and in varied contexts such as education or professional training [2–4]. STEM and computer science careers are on the rise, with strong demand in the job market. Serious games are often applied to increase interest in these disciplines (i.e., Science, Technology, Engineering, and Mathematics) by making learning more interactive and engaging for the broader public.

Moreover, despite the high demand and promising career opportunities, the gender representation in such careers remains highly imbalanced. According to UNESCO, in 2021, only 28% of engineering graduates and 40% of computer science graduates were female [5]. For instance, in Europe, women represent 41% of all science and engineering employment, and this percentage is even lower in computer science and tech roles [6,7]. With this scenario of high demand and gender imbalance, computer science is a field of interest [8], particularly considering the perceived potential of games to teach programming and make it more accessible and attractive to different audiences, such as school-aged girls.

The growing interest in introducing and improving computational thinking and programming knowledge among young people is evident in initiatives from the OECD's Programme for International Student Assessment (PISA). Computer science has been promoted as a new core subject, as reflected in the fact that the new PISA assessment for 2025 will include a new test called "Learning in the Digital World". This test will evaluate students' capacity to engage in an iterative process of knowledge-building and problem-solving using computational tools [9]. Based on the published information, it seems that this test will include questions to be solved using a visual block-based programming language (similar to Scratch). Therefore, schools will need to improve their coding teaching process, and teachers will require validated tools that help them teach programming and enhance the computational thinking skills of students.

There are several examples of games used to teach science (Physics Playground [10]), technology, chemistry (Minecraft Education [11]), engineering (SimCityEDU [12]), and mathematics (DragonBox [13,14]). As some studies have shown, serious games can also be effective tools for teaching programming [15], and playing games can attract new audiences to STEM careers [16]; however, for serious games to be effective and suitable in real-world classroom environments, significant barriers must be addressed. For instance, it is necessary to prove the effectiveness of each serious game and to provide teachers with tools to monitor players' learning. This includes understanding the learning process and identifying the issues and needs of students, both during and after gaming sessions.

This study explores the use of Game Learning Analytics (GLA) with a programming learning game. GLA involves the collection, analysis, and visualization of data gathered from players' interactions in a serious game. The purpose is to gain insights for multiple purposes (game validation and player assessment) and data that may be of interest to multiple stakeholders (game designers and developers, students, and teachers) [17]. Traditionally, the validation of educational games and their design were based on experiments including pre- and post-questionnaires or capturing videos of the game sessions, which then needed to be analyzed to identify key moments. With the emergence of e-learning, MOOCs, and the need to track many students, learning analytics techniques have begun to be used. These techniques can now also be applied to games (i.e., GLA) by collecting data on player interactions. The use of GLA allows for the analysis of game sessions in a faster and more effective way than video analysis, although it has the limitation that interactions outside the game in the real environment during the session are not captured. On the other hand, the large amount of data that can be collected on player interactions and the possibility of combining them with data from questionnaires allow for more advanced analysis to determine behavioral patterns or even develop predictive models.

In this paper, we discuss how we used GLA to identify the barriers and validate the game design of a serious game aimed at supporting teachers in programming classes. The game introduces basic programming concepts to teenagers and evaluates players' behavior and learning, helping to understand the key issues faced during gameplay. The main outcome of this validation approach is determining whether players have any issues in completing the game's levels or applying programming concepts, as well as gathering their feedback to improve future iterations of the game's design, which we intend to make openly and freely accessible.

The rest of this paper is organized as follows: Section 2 presents related work on programming learning platforms, educational games, and learning analytics; Section 3 describes the materials and methods used in this study, including the serious game designed to teach programming, the participants who played it, and how data were collected and analyzed; Section 4 presents the results of our study with respect to each of the research questions posed; Section 5 discusses the results; and Section 6 summarizes the main conclusions of our study.

2. Related Work

The current widespread demand for improving the learning of computational thinking and programming skills has fostered the development of many applications and online platforms to learn to program. There is a great variety of applications addressing different programming languages (and even only computational thinking concepts) covering different levels of knowledge and different age groups. For example, Codecademy and CodingGame are examples of complex platforms used to learn and practice languages such as C++, Java, or Python. We can also find virtual laboratories used in different fields of computer science, such as Java programming, cloud applications, or networks [18]. Furthermore, certain games, such as "Program your Robot" or "Mi superpoder es la programación", focus on improving computational thinking [19,20]. Although many games, platforms, and virtual laboratories to help with learning programming have been mentioned in the literature, most of them are not easily accessible or require a license payment [15,21].

At present, one of the most popular tools to learn programming in schools is Scratch, which allows students to develop their own games/projects by combining blocks. Scratch allows players to develop computational thinking skills and increases students' motivation, allowing them to quickly see the results of their code, thus driving their interest in continuing to learn about programming [22]. However, applying this open-ended tool in the classroom can be complex, as teachers must figure out how to adapt it to their needs. In particular, it can be complex to evaluate the knowledge acquired or to understand the problems that students run into while playing. Although initiatives such as Dr. Scratch [23] can provide some useful insights, open-ended tools are harder to deploy in classrooms than goal-oriented ones because, with sandbox-like games, it is up to students (or their teachers) to set goals, and the game cannot help to keep its players on target. More recent studies have also explored the use of serious games to teach advanced concepts such as parallel programming through a block-based visual language [24]. However, there remains a lack of free and open-source games for teachers to use in their classes when working on computational thinking and basic programming concepts. Specifically, games that provide sufficient insight for teachers to understand the players' progress, either during the game session (such that the teacher can act as a facilitator) or afterward (such that the teacher can reinforce some of the common mistakes or difficulties to improve the learning process) [25].

To provide evidence on the effectiveness of educational tools and study how users interact with them, Learning Analytics techniques can be applied to user interactions. Information gathered using these techniques can then help us to understand the learning process, classify user behaviors, and evaluate learning increments during the use of the tool [26]. However, reviews such as [27] have mentioned the need for more data-based research studies, especially at the high school and industry training levels. One of the problems that exist when collecting large amounts of data is the work required to preprocess them, as well as the selection and extraction of significant variables.

Using a standard format to gather user interaction data not only simplifies data collection but also streamlines later analysis of the collected data. The Experience API for Serious Games (xAPI-SG) profile allows for the collection of GLA interactions from serious games using an xAPI-based vocabulary that represents the most common interactions present in these games [28]. Each player interaction is captured as an xAPI-SG trace (a so-called statement), usually represented in JSON. Each xAPI statement contains three main fields, describing an actor, a verb, and the object of the action. Additional information can be included as extensions, and statements typically include a timestamp to capture the moment when the action occurred. The xAPI-SG standard defines a set of commonly used verbs and their associated types, including 'interacted' and 'used' to capture interactions with game objects, game items, and other characters (NPC or enemy); 'initialized', 'progressed', and 'completed' to capture the start, progress in, and completion of the whole serious-game or significant part thereof (e.g., within a level); and 'accessed' to capture changes in the screen or area. The use of a standard that is widely used in the educational field, such as xAPI, to represent the data allows for interoperability with other educational systems, such as Learning Record Storages (LRSs) or joint analysis of the data with other educational activities, such as those in Learning Management Systems (LMS). Working with a standard data format such as xAPI simplifies the creation of open analytics systems for a series of video games without beginning from scratch every time. Therefore, xAPI can ensure reproducibility and simplify different analyses of the collected information. Moreover, with our xAPI framework, aspects such as data ownership and privacy are guaranteed (complying with EU GDPR data privacy regulation by pseudoanonymization of the captured data).

Despite the increasing presence of games applied in educational scenarios, further research needs to be carried out to provide evidence on how such games can be evaluated through user interactions, as well as studying what types of tools and information can—when provided to teachers using these games—facilitate and simplify their use in the classroom.

3. Materials and Methods

This study explores the application of GLA to data collected from an open-source serious game, Articoding, when it was used in educational environments. Articoding is a serious game designed to teach basic programming concepts. The collection of user interactions and the application of analytics in the game has two main goals: (1) to validate whether the game's design is adequate to meet its educational objectives, and (2) to verify the extent to which players learn and determine any difficulties that they may have. With these goals, we aim to provide a case study contributing to the field of studies and use cases of GLA, while also providing a serious game that serves as a tool to teach programming that has been studied with real users and is available in an open-source manner. The latter will allow other studies to compare Articoding with other tools and to make modifications, both in the game design itself and in the interactions collected, in order to study such changes.

For this study, two Articoding game sessions were carried out in a high school in Madrid. While playing the game, user interaction data were collected. The players also filled out a questionnaire before and after playing the game.

3.1. Research Questions

The following research questions were proposed to guide this study regarding the use of GLA applied to a serious programming game:

RQ1. Can the applicability of the game in the classroom be measured using GLA? Can problems in the game deployment be detected?

RQ2. Can GLA information help to evaluate and find the problems and limitations associated with the game's design?

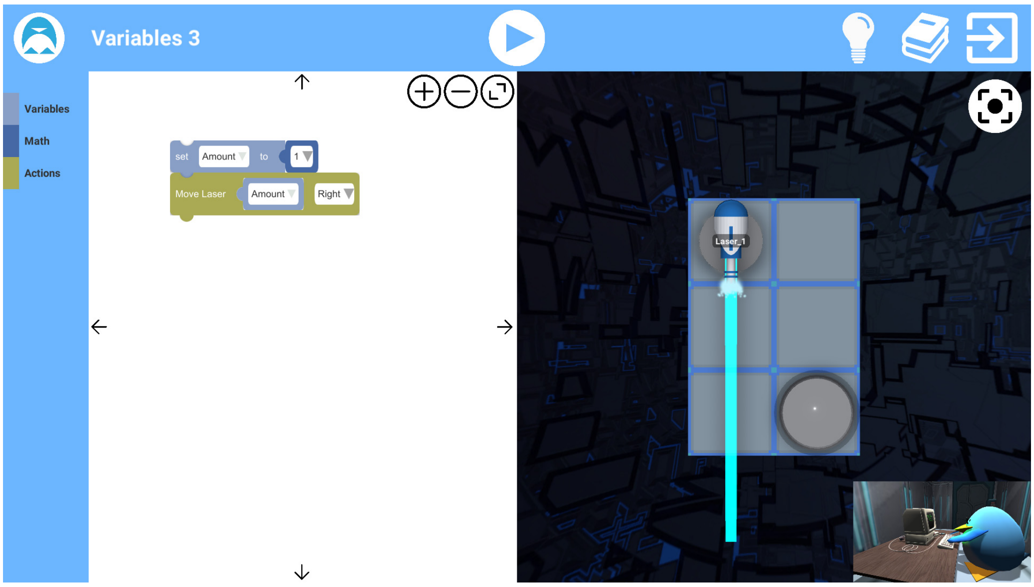
RQ3. Can the learning and application of the expected programming concepts by players be measured?

RQ4. Can engagement be measured using the collected interaction data? Do players like the game?

Finally, we propose to discuss the usefulness of GLA, its advantages over traditional assessment methods, and how it can encourage and facilitate the application of games in the classroom.

3.2. The Game Articoding

Articoding is a serious game aimed at teaching basic programming concepts and promoting Computational Thinking in students between the ages of 12 and 16, who have little or no previous programming knowledge. In the game, players must overcome levels by solving problems posed in a board-shaped scenario. In each level, the player's goal is to guide one or more laser beams to their targets using visual block programming. Players can activate, move, or rotate lasers to avoid obstacles. The laser should stay on the board and there are mirrors that can be rotated to guide the laser beam. When all targets are reached by their lasers, the player can advance to the next level. Figure 1 shows an example level in Articoding: the left side contains code blocks that can be modified and tested (executed), where the results can be observed on the game board to the right. At the top of the screen, a "play" button allows players to test their programs; meanwhile, in the top-right, players can request hints (bulb icon), access block descriptions (stack-of-books icon), and exit the level.

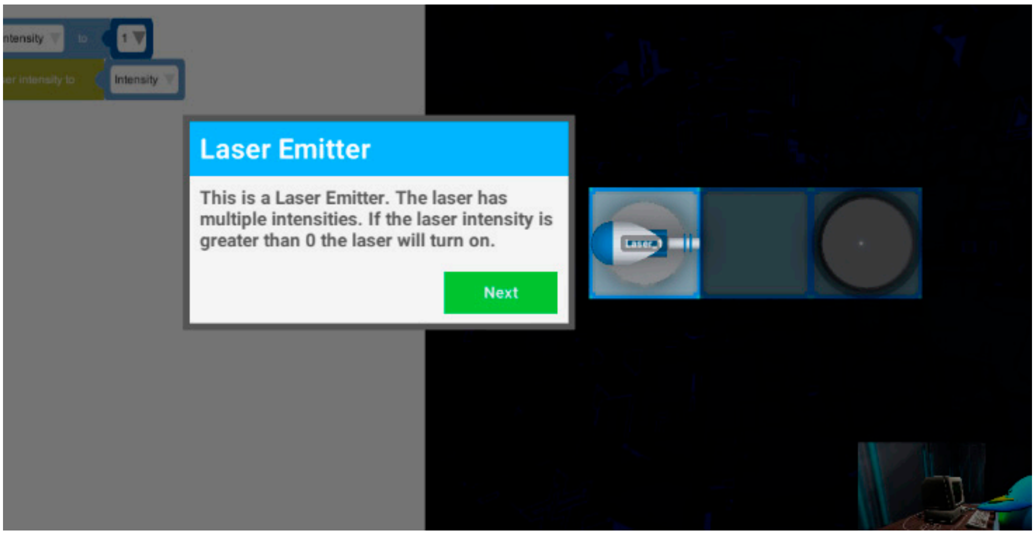


```
set Amount to 1
Move Laser Amount Right
```

Figure 1. Screenshot of a level in the game Articoding.

The game's levels are grouped into five categories, which correspond to programming concepts to be learned through the game: Variables, Data Types, Basic Operators, Loops, and Conditionals. Each category has several levels to be completed; in particular, all categories contain 7 levels, except for the Loops category which contains 9 levels. The categories must be played one after another in a fixed sequence, starting with Variables and ending with Conditionals. Within each category, the levels must also be played in their corresponding order. Figure 1 shows the third level in the first category (Variables).

A tutorial is provided in each new category or when a new element is introduced in the game, in order to show players how they are supposed to interact with it; for instance, Figure 2 shows an example tutorial explaining how in-game lasers work and how to activate them. Each game level also includes a hint system to help students who get stuck: a button shaped like a lightbulb is shown at the top-right of the level interface which, when pressed, draws arrows on the board indicating how board elements can be moved to solve the current puzzle. Up to a maximum of three different hints can be requested for each level.



Laser Emitter

This is a Laser Emitter. The laser has multiple intensities. If the laser intensity is greater than 0 the laser will turn on.

Figure 2. Partial screenshot of a tutorial example in Articoding.

For the learning process, in [29], programming problem solving was divided into four steps: (1) understand the problem; (2) determine how to solve the problem; (3) translate the solution into a computer language-based program; and (4) test and debug the program. However, students had difficulties expressing the solutions or developing instructions in the way that computers could execute them [30]; therefore, novice programmers need strategic problem-solving knowledge beyond syntactic knowledge [31].

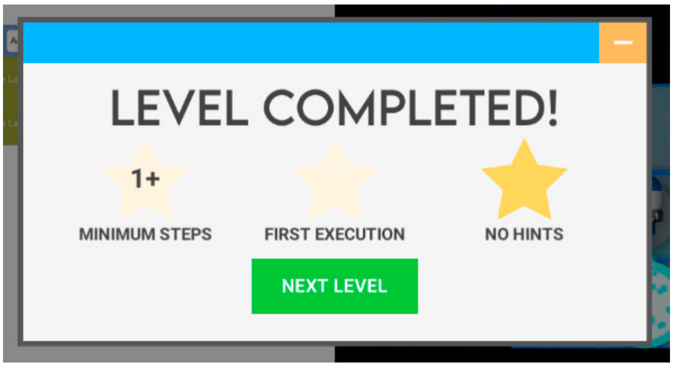
Articoding allows players to work through the four steps in problem-solving: (1) understanding the objective of the level; (2) finding the steps that the different elements of the game must follow in a level to reach the objective state; (3) translating their solution idea to the proposed language (Blockly); and, finally, in case of failure, (4) identifying the error and retracing their steps to solve it. Moreover, the simplicity of blocky and its proximity to natural language allows us to abstract the syntactic characteristics of programming languages and facilitate the translation of the player's solution into concrete instructions for the machine using basic programming concepts. This approach—namely, through problem-solving and not only teaching a programming language—is what allows for the development of computational thinking, which authors identify in terms of the processes of decomposition, abstraction, algorithmic design, debugging, iteration, and generalization [32].

Finally, to further engage players, a reward system provides up to three stars depending on the player's performance when solving each level. This is designed to get students to think through their answers, reflect on their solutions, and seek to achieve optimal solutions in terms of board movements. Stars are awarded for:

- Completing a level without using hints.

- Completing a level on the first run.
- Completing a level with the minimum number of steps, in terms of actions that modify the state of the board (e.g., moving or rotating an element, or changing the status of a door).

For instance, in the scenario shown in Figure 3, the player has received the "no hints" star but spent one step more than necessary (hence the 1+ and greyed-out minimum-steps star), and failed to solve the level on their first try. Figure 4 shows the complete flowchart of Articoding screens.



LEVEL COMPLETED!

1+

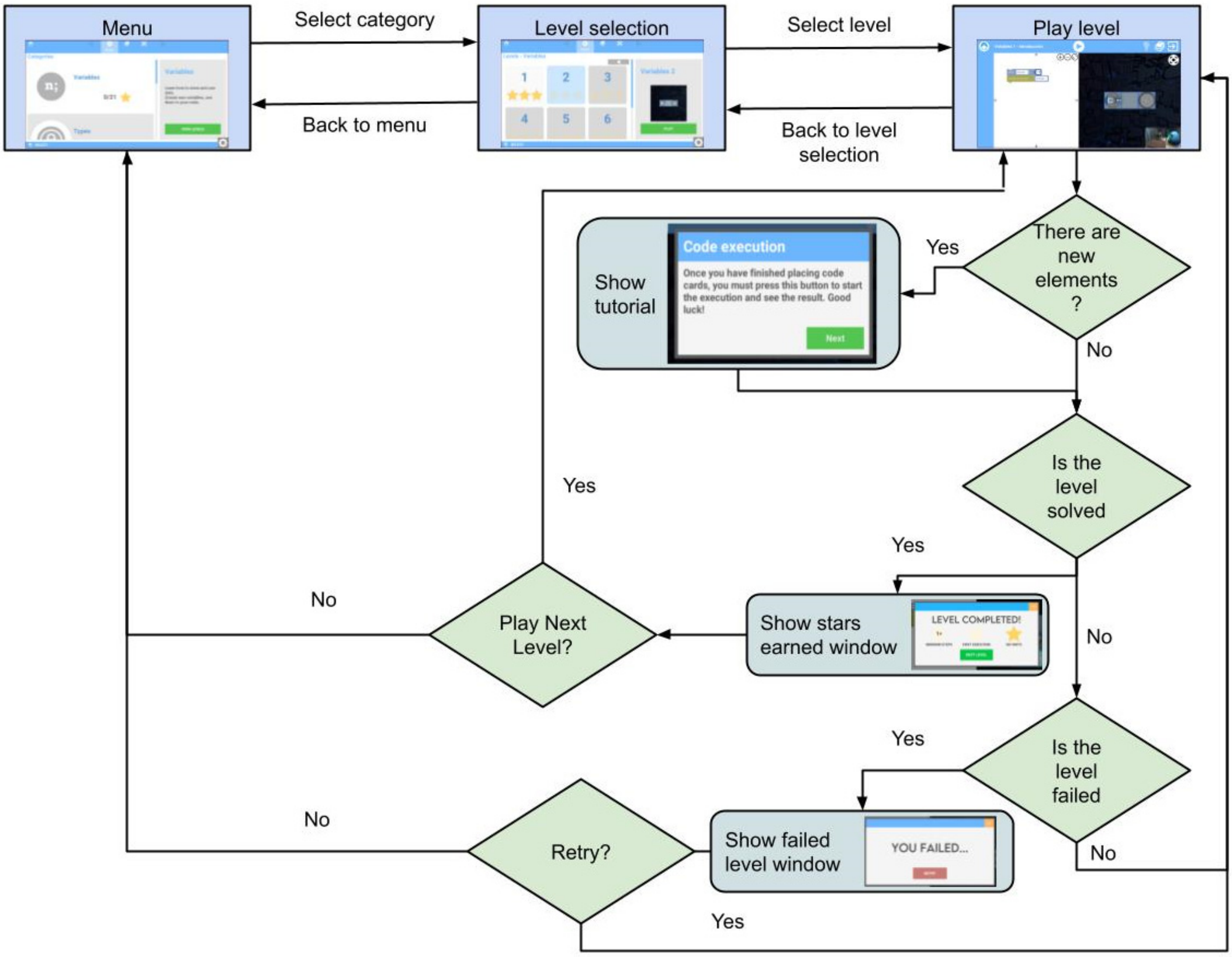
MINIMUM STEPS

FIRST EXECUTION

NO HINTS

NEXT LEVEL

Figure 3. Partial screenshot of an example of stars obtained after solving a game level in Articoding.



Once you have finished placing code cards, you must press this button to start the execution and see the result. Good luck!

Figure 4. Articoding screens flowchart.

The interest in using this game to study the use of GLA is due to its complexity, the variety of interactions that can be performed by the user, the playtime, and its replayability. In addition, the game is open-source (Github repository: https://github.com/e-ucm/Articoding23-24; accessed on 20 March 2025), which will allow other researchers to contrast and use the game as well and address a topic of social interest.

3.3. Game Sessions and Participants

To put into practice the advantages of GLA applied to serious games, we applied it to Articoding with 134 students from a high school in Madrid, Spain. The study was performed as an educational activity in the school. This activity was accepted by the school principal, who oversaw describing the activity and its characteristics to the parents of the students. Both the students and the school were informed of how the data would be collected, used, and managed prior to the experiment.

Participants completed two 50-min sessions on different days. In addition to playing the game, students completed questionnaires to gather their (self-reported) previous programming knowledge and, after playing, their opinions about the game. The questionnaires included the following questions:

- Pre-test
  - Demographics: age and gender.
  - Use of videogames: number of days and hours that they play videogames, and platforms used to play them.
  - Use of videogames in class: whether they had previously used games in class.
  - Programming knowledge: previous knowledge about programming (e.g., with Scratch) and programming courses previously taken.
- Post-test
  - Six 5-point Likert questions about perceived ease of use.
  - One 5-point Likert question about perceived usefulness.
  - Three 5-point Likert questions about attitude and intention towards use.
  - Four free-text questions to provide any other opinions of the game. Questions ask about positive and negative characteristics of the game, learning perceptions, and the option of adding a level editor to the game.

Of the participants, all session data (pre-post questionnaires and interactions) were successfully captured for 114 students; 20 students either experienced technical problems or did not fill out the post-experiment questionnaire. Excluding 3 participants who chose to not report their gender, students were balanced by gender (47% female, 53% male). Figure 5 shows the distribution of age and gender of the final set of participants. Of the 114 valid students, 91 played two sessions and 23 played only one game session.

Regarding the previous experience with programming, just 15 students answered not know or had used Scratch or other languages, while 15 were not sure. On the other hand, 69 students indicated that they had attended some type of programming course or class; in particular, the school where the experiment was carried out teaches optional subjects in which Scratch is used.

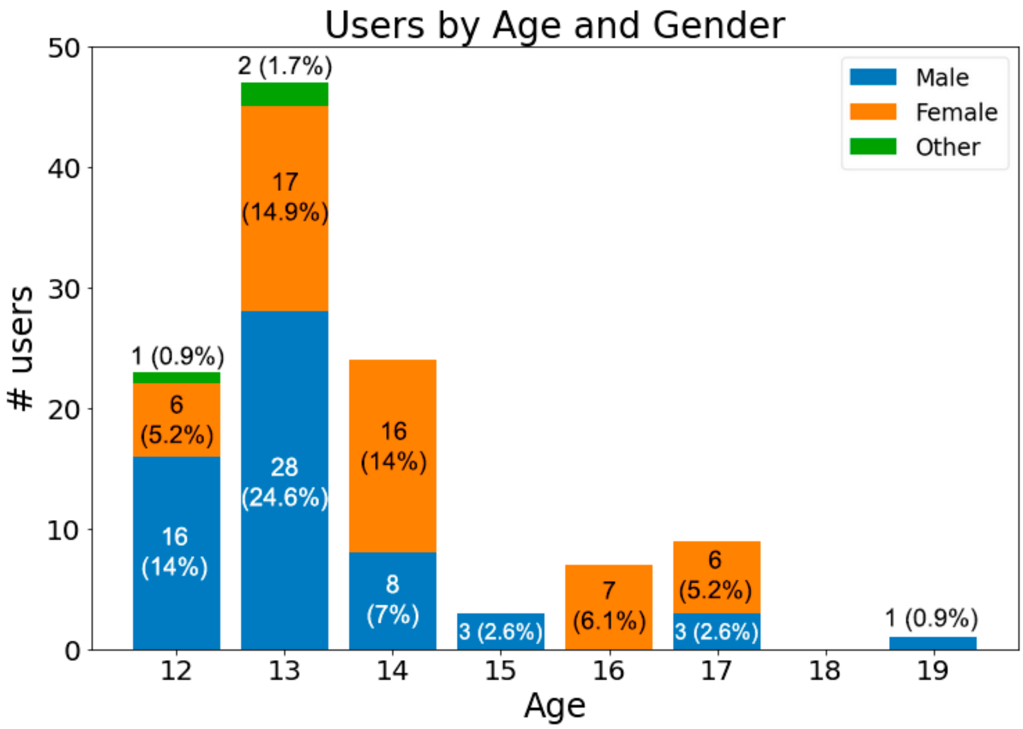


Figure 5. Age and gender distribution of participants in the study.

3.4. Interaction Data Collection

Articoding integrates an analytics system that allows player interactions to be captured and collected. It can be configured to send collected interactions either in real-time or after the game session is completed. The data collection process was based on the use of an xAPI-supported tracker and the analytics environment of the e-UCM research group [33,34]. The tracker collects user interaction data and sends that data to the server in real-time following the xAPI format. The analytics environment also simplifies the planning and running of experiments using serious games, handling—among other tasks—player authentication, pre- and post-game online questionnaire management, and allowing for easy retrieval of questionnaire results together with collected interaction data linked to each player. Data collection is pseudo-anonymized at the source by providing unique, random codes to each player, which allows questionnaires to be linked to analytics information without revealing the identities of players. This avoids storing identifiable information of students together with their responses, while still allowing (pseudonymous) use of analytics.

During the game sessions, a total of 98,704 unique traces were captured from all participants, which translates to more than 700 traces per player. Such traces included all interactions with game elements, together with any significant changes to each player's game state. The most meaningful trace types for our analyses are described in Table 1. These traces capture the start, progress, and completion of game levels, including whether the level was successfully completed, the number of steps needed to complete the game level, and the stars obtained when doing so (given by the three fields with the extensions minimum_steps, no_hints, and first_execution).

As an example, Figure 6 displays an xAPI-SG statement collected from Articoding representing that the actor (with anonymous identifier 'fybu') completed the game level 'types_6' (sixth level of the Data Types category) at the given timestamp. It was completed in 5 steps, obtaining all 3 possible stars, and therefore a (maximum) score of 3 stars was obtained.

Table 1. Most meaningful traces collected from Articoding.

Verb	Object ID	Additional Fields Used as xAPI Statements' Templates	Meaning
accessed	game level [1]	object.definition.type = level	Access to game level
initialized	videogame	object.definition.type = serious-game	Start of full game
	game level [1]	object.definition.type = level	Start of game level attempt
completed	game level [1]	object.definition.type = level result.success = true result.score (Number) result.extensions.articoding://minimum_steps (Bool) result.extensions.articoding://no_hints (Bool) result.extensions.articoding://first_execution (Bool) result.extensions.articoding://steps (Number)	Successful completion of game level

	game level [1]	object.definition.type = level
		result.success = false
			Unsuccessful completion of game level

[1] Possible game level values correspond to levels in each level category. More specifically: variables_X: X in [1,7]; types_X: X in [1,7]; basic_operators_X: X in [1,7]; loops_X: X in [1,9]; conditionals_X: X in [1,7].

```
{
“actor”: {
“name”: “fybu”,
“account”: {
“name”: “fybu”,
“homePage”: “https://analytics-api.es”
}
},
“result”: {
“score”: {
“raw”: 3
},
“extensions”: {
“articoding://minimum_steps”: true,
“articoding://no_hint”: true,
“articoding://first_execution”: true,
“articoding://steps”: 5
},
“success”: true
},
“verb”: {
“id”: “http://adlnet.gov/expapi/verbs/completed”
},
“id”: “5806c959-39bd-4688-9ec3-7d3d8ad8f219”,
“object”: {
“definition”: {
“type”: “https://w3id.org/xapi/seriousgames/activity-types/level”
},
“id”: “https://analytics-api.es/activities/60c36c/types_6”
},
“timestamp”: “2021-06-17T07:27:58.276Z”
}
```

Figure 6. xAPI-SG statement captured from Articoding representing a Data Type level completed with three stars.

All these collected data allowed us to apply GLA. Analyses can be applied in near real-time to the data being generated or at the end of the game sessions. With the collected data, we can also reproduce the players' interactions with the interface elements and their way of solving the game levels and using programming blocks.

4. Results

In this section, we present the results obtained from the data collected from players' interactions and questionnaires corresponding to each of the stated research questions.

4.1. RQ1. Can the Applicability of the Game in the Classroom Be Measured Using GLA? Can Problems in the Game Deployment Be Detected?

Through analyzing the collected interaction data, we can determine whether a player had not played or had played in several different sessions, as well as for how long. However, it is much more complex to understand why a player has not played the game or stopped playing before the session ends. For the latter, data external to the game itself are required.

Analyzing the student interaction data, 130 users played the game, and interaction data were obtained. There were 4 students without interaction data, for reasons unknown. Most likely, there were problems with the computer connection or they were playing with a colleague. Of the 130 students who had interaction data, 14 did not attend the second planned game session and, so, did not complete the final questionnaire; furthermore, one of the students started playing very late and only played for 13 min. Another 2 players left the session without completing the final questionnaires. As a result, 85% of the players completed the game sessions satisfactorily, carrying out the proposed activities.

The data revealed that most users were able to play without any problems associated with the operation or design of the game. Although some did not complete the final questionnaire, there were four players from whom we did not obtain any interaction data. Despite this, the game could generally be applied satisfactorily and the collection of interaction data allowed us to filter out those players who did not interact with the game during a large part of the session. However, we cannot know the reason why these players did not play or did not play long enough, due to these being associated with events external to the game, which were therefore not captured. Table 2 shows the number of participants who completed each phase of the study.

Table 2. Total of participants by level of activity completion and data collected.

Level of Completion	Number of Participants	Description
Total participants	134	The total number of students who played the video game.
Valid participants	114 (59 males, 52 females, 3 others)	The total number of students who played the video game, sent all questionnaire data, and for which interaction data were obtained.
Participants who played two sessions	91 (53 males, 35 females, 3 others)	The number of students who played two 50-min sessions.
Participants who played one session	23 (6 males, 17 females)	The number of students who played just one 50-min session.
Participants who completed the "Variables" category	106 (57 males, 46 females, 3 others)	The total number of students who completed all levels for the first category ("Variables").
Participants who completed the "Types" category	26 (15 males, 10 females, 1 other)	The total number of students who completed all levels for the second category ("Data Types").

4.2. RQ2. Can GLA Information Help to Evaluate and Find the Problems and Limitations Associated with the Game's Design?

With the interaction data, it is possible to know exactly how many levels each player completed, as well as how much time and attempts they needed. It is also possible to analyze the types of interactions that were performed with the different code blocks and what solution was reached for each level. If all players spend too much time on a level (or do not even complete it), it indicates problems in the design of the level. However, if this happens for only one player, it may indicate that this student has problems with the concept of the level. On the other hand, too much variation in the number of attempts and time needed by all players between levels will indicate that the growth in difficulty is not adequate.

A total of 91 players finished two 50-min sessions and completed 12 levels on average, with a minimum number of 5 completed levels, and a maximum of 18 (23 students only played a single session and, therefore, are not counted in this respect). Players completed each level within an average of 6.4 min. In the two sessions, only 26 users were able to fully complete the first two categories: Variables and Data Types. No user was able to complete the third category, Basic Operators. The interactions show that no further progress was made due to a lack of time, and not because there were problems with the game levels.

Regarding the progress of the levels and their completion, the 53 male players that finished two 50-min sessions were able to complete 12 levels on average (minimum of 5, maximum of 18). In comparison, the 35 female players who finished two 50-min sessions completed 11 levels on average (minimum of 6, maximum of 16). The completion of levels followed a normal distribution, and the t-test showed no statistically significant difference between the number of levels completed by male and female players (p = 0.29).

Furthermore, the *t*-test indicated no significant differences between genders in playing time per level. The 53 male players who finished two 50-min sessions completed each level within an average of 6.3 min, while the 35 female players needed 6.5 min per level on average. Table 3 shows the time needed (in minutes) to complete the Variables and Data Types categories according to the different players' gender and course.

Table 3. Time (in minutes) to complete the Variables and Data Types categories by gender and course of players.

Category	Player Characteristics		Number of Users	Mean	Standard Deviation
Variables	Gender	Male	57	24.36	7.34
		Female	46	25.35	8.30
	Course	First Year	49	25.52	8.24
		Second Year	44	24.88	7.48
		High School	13	21.55	5.59
Data Types	Gender	Male	15	39.57	7.24
		Female	10	47.43	11.49
	Course	First Year	11	42.80	5.65
		Second Year	14	44.15	10.35
		High School	1	19.10	-

To study the design of the game levels and determine whether they have the intended difficulty, the use of hints, the number of level retries, and the time needed to complete the game levels were analyzed. Too many players using hints or needing too many attempts at a single level may indicate that a level is more difficult than expected at that point in the game. Studying the time spent by players on each level is also important, as placing many levels that take players a long time together can cause them to become frustrated and lose the feeling of progress. It is important, as a matter of good game design, to keep the time needed to complete each level relatively short, even as the difficulty increases progressively and new concepts are introduced.

To determine whether the players had problems in completing certain levels, their use of hints was analyzed. Table 3 shows the number of clues used in total by players in each level, for the Variables category. The data indicate a roughly linear increase in the use of hints, except for levels 4 and 5. For the Data Types category, few hints were used overall, except for levels 2 and 3, which needed a high number.

Additionally, Table 4 shows the number of level retries by players required to complete each level, explicitly stating the mean, standard deviation, and maximum number of retries. For the Variables category, a higher number of retries was seen in level 3, while levels 2 and 3 of the Data Types category concentrated the highest number of retries in that category.

Table 4. Total number of hints used by players in each level per category.

Category	Level						
	1	**2**	**3**	**4**	**5**	**6**	**7**
Variables	0	0	6	11	7	14	16
Data Types	0	14	16	2	0	4	0

The data presented in Tables 4–6 indicate that the second and third levels of the Data Types category seem to be more complicated than should be expected at that stage of the game. The data also verified that the number of attempts and time required to complete a level is correlated with the number of blocks needed to obtain its solution, and is also correlated with the number of blocks provided as part of the solution to the user at the beginning of the level. It will be interesting to study how the use of loops affects this fact, as they allow players to make more movements with fewer code blocks.

Table 5. Average number of level retries by players for each level, per category, with standard deviation and maximum.

Category	Level						
	1	**2**	**3**	**4**	**5**	**6**	**7**
Variables	AVG = 1.06 SD = 0.33 max = 4	AVG = 2.23 SD = 1.62 max = 9	AVG = 3.37 SD = 3.13 max = 24	AVG = 2.77 SD = 2.04 max = 13	AVG = 4.21 SD = 2.69 max = 14	AVG = 5.08 SD = 5.01 max = 24	AVG = 2.90 SD = 2.18 max = 13
Data Types	AVG = 1.27 SD = 0.92 max = 6	AVG = 6.22 SD = 5.28 max = 28	AVG = 5.79 SD = 5.96 max = 36	AVG = 3.27 SD = 2.52 max = 15	AVG = 3.15 SD = 1.91 max = 9	AVG = 3.64 SD = 3.12 max = 15	AVG = 2.85 SD = 3.61 max = 19

Table 6. Average time (in minutes) to complete each level for the first time per category.

Category	Level						
	1	**2**	**3**	**4**	**5**	**6**	**7**
Variables	AVG = 2.04 SD = 1.06	AVG = 2.14 SD = 1.20	AVG = 2.83 SD = 2.30	AVG = 4.10 SD = 2.52	AVG = 4.22 SD = 1.92	AVG = 6.32 SD = 4.43	AVG = 3.64 SD = 2.05
Data Types	AVG = 0.91 SD = 0.95	AVG = 8.69 SD = 6.19	AVG = 10.10 SD = 5.28	AVG = 6.78 SD = 2.96	AVG = 6.07 SD = 2.46	AVG = 9.83 SD = 3.74	AVG = 6.98 SD = 3.27

Two of the questions in the post-test further assessed how players perceived the difficulty of levels. The statements of these Likert questions were "The game has frustrating levels" and "The difficulty grows appropriately". Figure 7 shows the participants' answers to these questions, where 5 indicates maximum agreement with each statement. While some pointed out that there were frustrating levels, most players considered that the difficulty in the game increases appropriately.

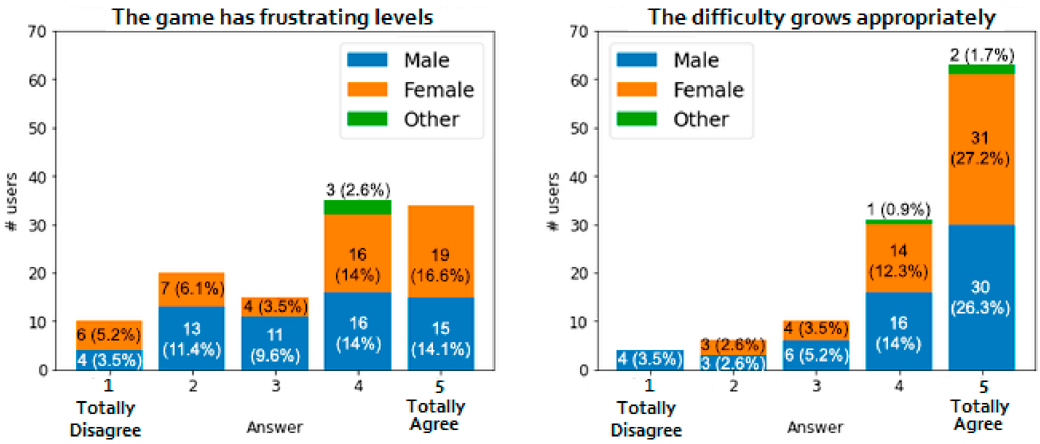


Figure 7. Participants' opinions on level design in Articoding levels.

Finally, no statistically significant differences were found between players who reported having attended programming classes and those who did not; however, the five players who completed the most levels did report having attended programming classes. The lack of further differences may be due to most of the student's prior knowledge of Scratch and the appropriate introduction to the game elements.

4.3. RQ3. Can the Learning and Application of the Expected Programming Concepts by Players Be Measured?

To determine the extent to which players learned programming with the game, the first step was defining a set of indirect measures of learning through the interactions with the game. This approach is commonly known as Stealth Assessment [10], where an evidence model needs to be initially defined to then be able to compare the information gathered from players' interactions and update the corresponding knowledge acquired. Another option is to use external questionnaires to assess knowledge before and after playing (e.g., Bebras or other instruments used to assess programming knowledge [35,36]); however, this increases the time required for the experimental sessions. In this scenario, players are considered to understand the concepts included in the game if they:

- Can progress in the game levels without having too much difficulty (i.e., making a low number of attempts).
- Do not spend too much time without knowing what to do in the game (i.e., without long periods of inactivity).

If the contrary situation occurs—that is, if a player is frequently inactive for long periods, or needs many attempts to complete the game's levels—the player is considered to have not successfully understood the concepts introduced in the game. Particularly, this study considers that players experience difficulties in a game level if they fail more than 3 attempts to complete the level; and players are considered inactive in a level when their inactivity time is an outlier in the distribution, considering such values as being above *Q3* + 1.5**IQR*. When a player has difficulties on 3 levels or is inactive on 3 levels of the same category, the player is considered to have not adequately learned the concepts and/or not understand what needs to be done in the game.

For the game levels related to the Variables category, there were 22 players with repeated difficulties (difficulty in more than 3 levels in the category) and 19 players with repeated inactivity (more than 3 levels in the category). Overall, 8 players satisfied both conditions (difficulties and inactivity), in the Variables category. For the Data Types category, there are 17 players with repeated difficulties and 10 players with repeated inactivity, while only 2 players satisfied both conditions (difficulties and inactivity) in this category.

The analysis also involved checking that the players who had problems learning the programming concepts included in the game (i.e., those who repeated levels too many times and were inactive for too long) stated in the post-test that they did not want to keep playing the game.

During the experiments, some players reported issues understanding the concept of "variable". These perceived issues during the experiments were contrasted with the collected data. This was clear with respect to some in-game behaviors; for instance, players declared variables only in the place where they needed to use their values, which may mean that they did not understand that they could be declared at the beginning of code and used at any time. There also seemed to be problems with understanding that a variable can be used for multiple purposes once it stores a value, as students sometimes created two variables with the same value; for example, to rotate and move the laser. The frequency with which the following behaviors occurred is shown in Table 7.

Table 7. Number of players with issues using variables in completed levels in the Variables and Data Types categories.

Level	Issue *								
	VV	OC	AC	NI	TV	SS	NV	SM	CC
Variables_1	1	1	1	0	0	0	0	0	0
Variables_2	2	33	33	2	0	0	29	0	18
Variables_3	3	6	6	5	0	0	2	0	0
Variables_4	1	0	0	1	0	0	1	1	0
Variables_5	4	66	10	4	0	51	58	3	52
Variables_6	2	40	18	3	0	30	34	2	37
Variables_7	3	59	18	5	2	2	5	2	57
Data Types_1	0	1	1	1	0	1	0	0	0
Data Types_2	3	67	7	4	2	4	3	1	61
Data Types_3	0	24	0	1	7	1	1	0	42
Data Types_4	0	6	0	5	5	0	1	1	25
Data Types_5	0	26	0	0	4	0	1	0	1
Data Types_6	2	24	0	2	2	3	1	2	0
Data Types_7	0	7	0	0	0	1	5	0	9

* Legend of issues found: VV—Player initialized a variable with itself, such as "move = move". OC—Player used a constant block instead of a variable at least once. AC—Player used only constant blocks instead of variables. NI—Player created a variable without assigning a value. TV—Player created more variables than necessary. SS—Player always initialized variables just before using them. NV—Player initialized a variable that was never used. SM—Player initialized more than one variable with the same value. CC—Player used the same constant value instead of creating a variable.

4.4. RQ4. Can Engagement Be Measured Using the Collected Interaction Data? Do Players Like the Game?

Besides the fact that the game successfully introduces players to basic programming concepts, we are also interested in maintaining player engagement, in order to ensure that they want to continue improving, attempting to complete the levels without hints or errors and in as few steps as possible. To measure the extent to which the game engages players, two aspects were analyzed:

- Players try to get as many stars as possible, retrying levels if necessary.
- Players do not remain idle for large amounts of time and continue to play throughout the session.

Player inactivity is also related to player learning (RQ3). A boring game that does not engage players will make them lose interest in advancing as fast as possible to find new challenges, causing them to not pay attention and miss out on learning or trying out new concepts. On the other hand, a player who does not learn the necessary programming concepts to continue advancing or does not understand the game will lose interest in the video game itself, thus performing fewer and fewer interactions.

Comparing players with two completed sessions according to their gender, less time passed between interactions on average for male players than for female players (Figure 8). This time difference was statistically significant with a moderate effect size (Mann–Whitney: $p < 0.001$; r = 0.47), which may point to a different way of playing (be more cautious) that does not translate into an increase in the number of attempts or the number of levels completed (Table 8).

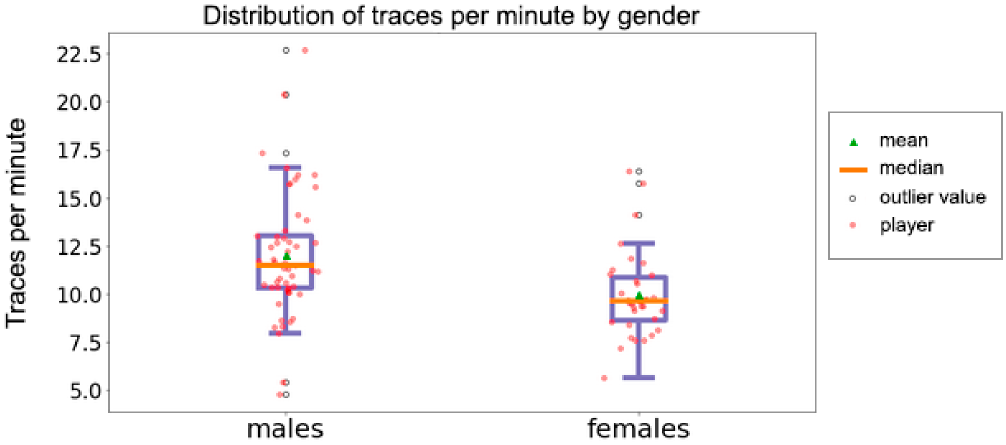


Figure 8. User interactions per minute by gender.

Table 8. Average traces and levels completed per minute by gender.

	Traces Per Minute	Levels Completed
Male	12.01; SD = 3.27	0.160; SD = 0.041
Female	9.94; SD = 2.23	0.154; SD = 0.027

Regarding the stars achieved, the number of players who achieved all three stars in a level decreased as the levels advanced in a category. Figures 9 and 10 show the number of players achieving 0, 1, 2, and 3 stars in each level of the Variables and Data Types categories, respectively. We can see, from Figure 9, that all players finished the first level with 3 stars (the maximum) while, in later levels, it was much rarer for players to get all 3 stars. It can be noted in Figure 10 that only a fraction of players finished the last level.

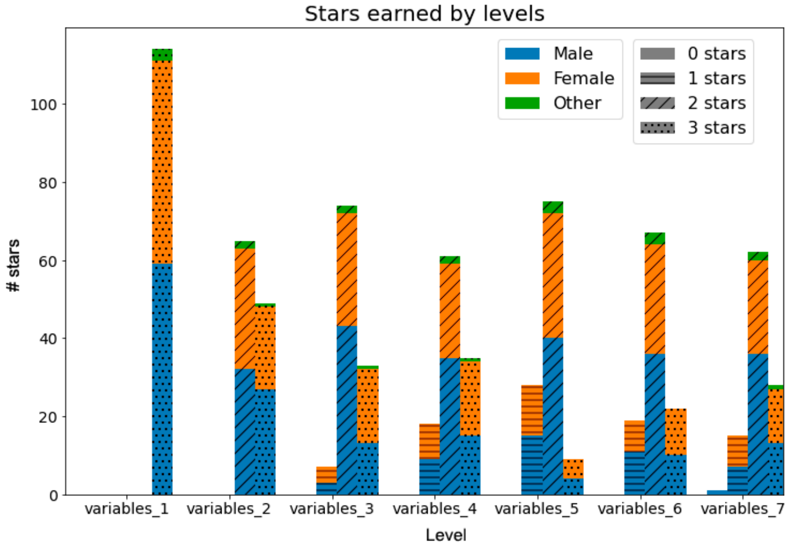


Figure 9. Awarded stars in "Variables" levels.

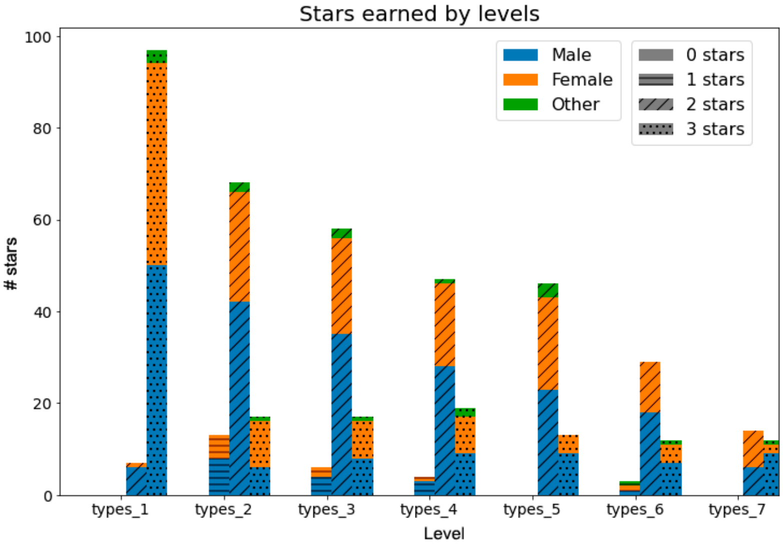


Figure 10. Awarded stars in "Data Types" levels.

Most players obtained 2 stars, avoiding using the clues in each level and trying to make as few moves as possible. The most difficult star to earn is the one for completing a level on the first attempt. As for the repetition of levels, 16 players replayed levels to get more stars, and 5 of those did it in multiple levels with the aim of getting 3 stars. Notably, 4 of the 16 players who repeated a level did not manage to improve. In addition, almost all retries occurred during the levels of the initial Variables category. The delineation by gender is additionally depicted in Figures 9 and 10, demonstrating no significant difference in the number of stars achieved between male and female players.

If a player is committed to the game, does not get bored, is active, and progresses, we can conclude that they like the game. In this case, the players were also asked for their opinions through a post-game questionnaire. Figure 11 displays the results of the post-test question "Would you like to continue playing the game?", which indicates that players generally liked the game (65% would continue playing). However, there seemed to be a slight gender difference, as boys liked the game more than girls (81% of boys vs. 46% of girls); furthermore, girls were more indifferent to the game (33% were indifferent vs. only 5% of boys). Additionally, most players found the game simple easy to use, entertaining, and interesting (Figure 12). In this case, no correlation was found between the opinion given of the game in the questionnaire and the data on inactivity, number of levels completed, or stars earned. This may be due to various reasons such as the activity being mandatory, or because there was little data on players who did not like the game.

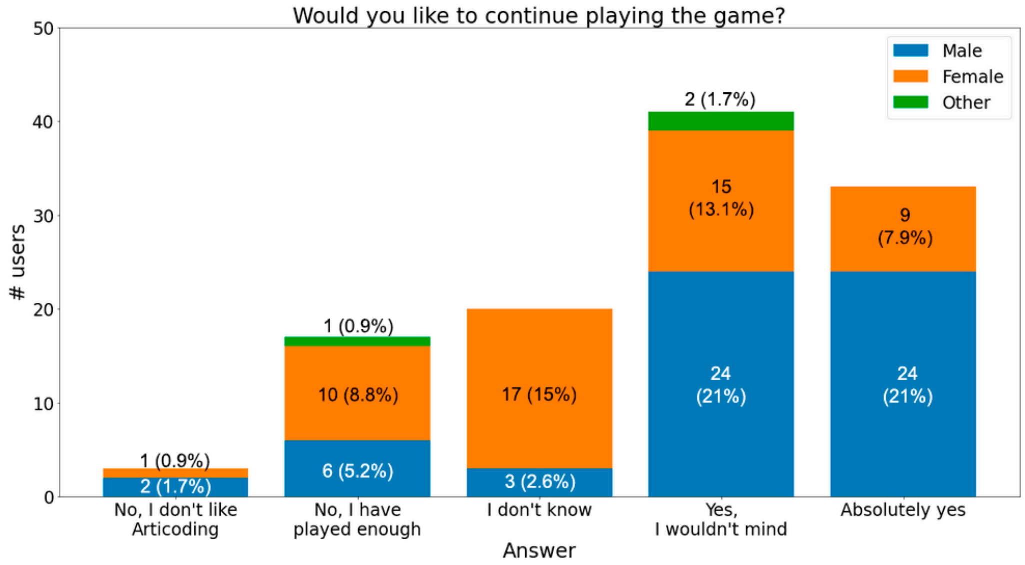


Figure 11. Opinions on willingness to continue to play by gender.

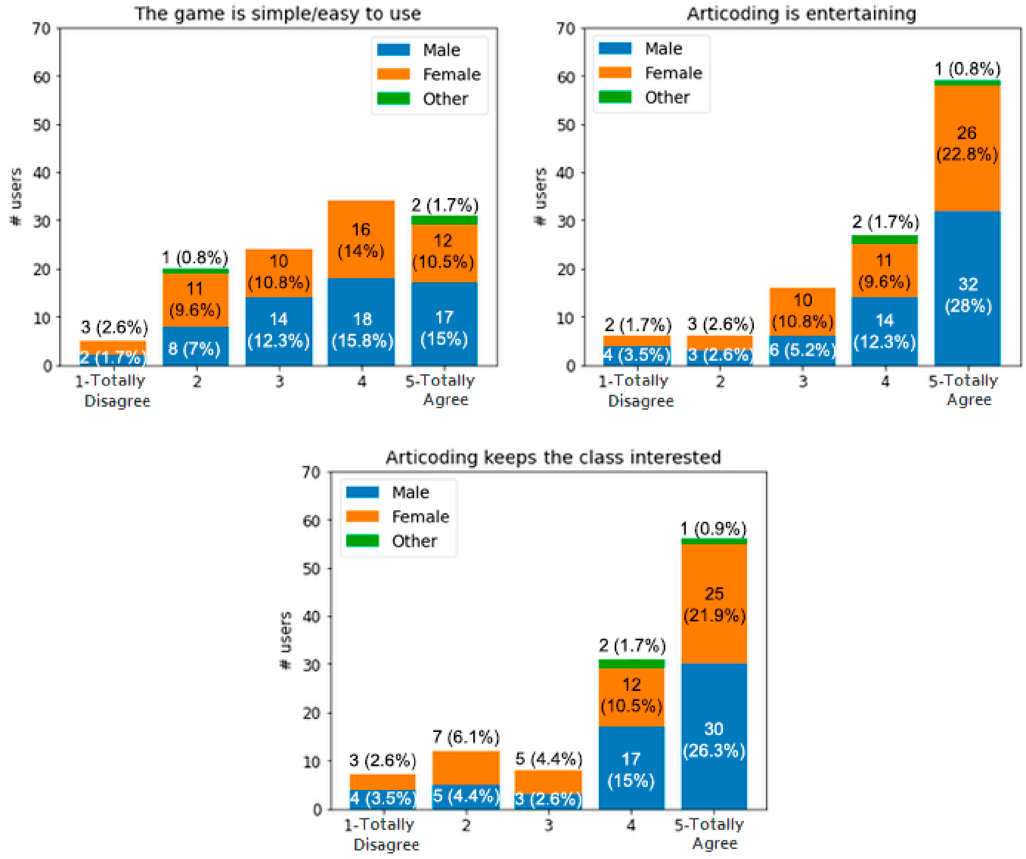


Figure 12. Opinions of players regarding whether the game is: easy to use, entertaining, and keeps the class interested.

5. Discussion

In this work, GLA allowed for an analysis of the behaviors of 114 students playing the serious game Articoding. Individual observation of each player in each game session together with an interview would allow us to better understand the students' learning levels and any problems that they may have; however, this would not be a scalable process and its cost, in terms of time or human resources, would be too high to implement in a real scenario. GLA therefore allowed us to study the learning progress of many more players and their behavior at a lower cost (in terms of both time and resources), compared to traditional methods. In addition, GLA allowed us to obtain more accurate information and, with a larger sample of users and, consequently, data to compare, we could better study the effects of the game by type of user and detect students who stand out in the way they play.

5.1. Applicability of the Game

With the collected data, we could determine how long the players played and how long they were inactive. GLA allowed us to study the time required for each level by the player and, therefore, better adapt the game to the needs of the target audience and the context of use; in this case, the students and teachers and the use of the game in the classroom, respectively. Considering the original design of the game and the characteristics of the experiments carried out (duration and number of sessions) we, as game designers, hypothesized that players would be able to complete at least the first three categories. However, the data revealed that no player could complete the third category (Basic Operators) and only 25 users were able to complete the second category (Data Types). Based on these results, the application of the game in the classroom context—where time is typically limited—needs to be modified: either the game would need to be used in more sessions to cover all included concepts, or the game design would need to be modified to include fewer levels and/or simplify existing ones, allowing players to advance further in the game. In future experiments, we hope to be able to study the time required for players to complete the remaining categories (Basic Operators, Loops, and Conditionals). This will allow us to more precisely adjust the game's levels, as well as the time required to complete the full game. With such information, we aim to additionally create a guide to help teachers apply Articoding as a more effective educational tool in their courses.

Moreover, with the current data, it may be necessary to dedicate a 1 h session for each category. However, we also have to take into account that there can be large differences in completion time between different players, and features should be implemented such that the fastest players can remain engaged while their slower peers finish; two options include optional levels and increased replayability.

However, we obtained no information about those players for whom no interaction traces were received but who filled out the initial questionnaire as they were present. Did they not come to play because they were late for class? After filling out the questionnaire, did they start playing with a classmate? Did they have to leave class for some reason?

5.2. Game Issues and Limitations

As a game design issue, the students commented that the clues did not help much. Possibly due to the fact that a potential star was lost when players asked for hints, the hint system was rarely used. Therefore, this aspect may benefit from being revised, both to provide more useful help and as part of a deeper overhaul to re-design how and when stars are awarded in future game versions. The use of GLA can help to compare different versions of the game in the future; for example, by studying the effects of giving more specific hints or providing guided levels. This may yield interesting results, as researchers generally do not agree about the effects of providing guidance to players [37].

Data and GLA pointed out another design issue regarding the difficulty of levels 2 and 3 in the Data Type category, as players had more trouble with these levels than in the final levels of this category, using more clues and requiring more attempts and time to solve them. As such, these two levels need to be simplified or moved to the later stages of the category, after simpler levels have been successfully solved. Finally, the analysis revealed that not many players replayed the levels to try to improve. It will be necessary to study whether this is because completing the levels with all the stars does not motivate players or if, due to the time limits, players prefer to advance further rather than get more stars. These issues were not reported by players, and many of the students were satisfied with the level of difficulty and their progress.

5.3. Student Learning

Measuring learning when playing a videogame can be complex, as we do not know the extent to which players can later apply the knowledge acquired in-game to real scenarios, unless the game design itself is focused on measuring that learning via in-game assessment [38] or stealth assessment [39]. In our case, the variables chosen to measure how the player improves and understands the concepts presented in the game were inactivity and difficulty (attempts) in completing the levels. However, these measures are indirect and non-standard; therefore, in subsequent experimental research, it may be necessary to add external tools to measure the knowledge gained by players during game sessions in order to validate the effectiveness of the game as a learning tool. For example, we could add a validated post-test to assess their learning. In such a case, it would also be interesting to compare the learning results determined through that external measure with those obtained via our indirect measures. It should not be forgotten that this game is intended to be a complementary and motivational tool that teachers can use to learn and practice basic programming concepts and to work on improving computational thinking; it is not intended to be a one-stop, self-contained programming tutor.

It is also necessary to emphasize that analytics may not capture all aspects of the game session. In this experiment, all players advanced through the levels at differing rates, and they had the game open during the game sessions. However, Learning Analytics does not capture cases where the teachers or researchers in charge of the session stepped in to address their doubts or help players complete a game level. Neither can we know exactly what happened to those players who appeared inactive: did they go to the bathroom? Did they interact with their classmates? There are studies that have shown, for example, how interactions with teachers or peers can be correlated with knowledge and computational thinking skills [40]. The process and analysis can be improved by adding more data sources, allowing for the application of multi-modal learning analytics. In this case, we complemented the interaction data with data collected through questionnaires; however, a multi-modal analytics approach would allow us to integrate even more data sources.

5.4. Engagement

The designed game maintained the interest of most of its users—at least in the sense of keeping idle times low—showcasing the engagement typically related to game-based learning and gamification [41]. This engagement was further contrasted with a question in the post-test, which indicated that 65% of the players would continue playing while only 17.5% would not play anymore. The other 17.5% did not know whether they would continue to play if they had the possibility.

For those players who were not engaged in the game, additional research would be needed to determine if they were not engaged due to the game mechanics and the game design, or if they did not understand the game or considered it to be too difficult.

However, the game was overall very well received by players, with most players enjoying the game (75.5%) and finding it easy to use (57%). These results align with previous research demonstrating the interest of students in using videogame in class in general, and in learning programming concepts in particular [42]. The advantages of using games to increase students' motivation before they have taken any computer science or programming courses have also been highlighted [43,44].

However, many players also pointed out that the game had levels that could be too frustrating (60.5%). As these levels may cause some players to give up or lose interest in the game, it is important to address this issue by revising those difficult levels and making the difficulty reasonably progressive in future versions of the game.

6. Conclusions

The use of Learning Analytics in the context of educational games has many benefits when attempting to understand the learning of players and detect the problems they encounter, ultimately allowing for the improvement of a game and its educational design. GLA can be oriented to both a game's design and development phase and its actual application in real environments, reducing the cost of obtaining relevant information regarding the use of the game.

In this study, GLA helped us to verify that the serious game Articoding is a useful tool to introduce programming concepts in an entertaining and interesting way to students. The use of analytics provided evidence that helped us to identify several educational issues and game design problems that still need to be addressed in future versions of the game. It also allowed us to identify problems associated with how players understand programming concepts such as variables. Some of these analytics could be further applied in (near) real-time, keeping teachers informed of the players' progress while the games are being played in their classrooms. This would allow teachers to better understand the concepts that their students find challenging, identify students who are struggling and need help, report on their progress, or maintain class control by highlighting those players who are inactive or interact little with the game.

Collected analytics information provides abundant evidence to drive an education-focused re-design of the game. Considering all these observations, we plan to develop a new version of the game that better explains the concept of variables, as well as re-designing several levels and improving the star system and the hints provided for each level. For example, evidence suggests that the in-game hints system is not useful, as players make very little use of them. In terms of levels, the data analysis showed a pronounced variation in difficulty between some of the levels, breaking the ideally increasing difficulty curve.

The Articoding game itself, as a free software project, is also the result of this work. Its open-source nature allows other researchers to easily build on this work and modify it. In addition, the use of the xAPI standard for data collection allows other researchers to create their own tools and data analysis platforms for Articoding, or to integrate it with existing tools that implement this standard.

With respect to the analysis, we also want to design more precise learning measures; for instance, establishing a measure of level difficulty, and comparing the improvement of players in levels that are equivalent in difficulty (i.e., levels that do not introduce new concepts and start with the same number of blocks provided).

An issue present in our results is that there was information that was not captured by the interaction data, which may be required to complement the analytics. Events outside the game, such as interactions between players or times when teachers or researchers helped players, directly affect their results and should be considered in the analysis. This highlights the importance of taking notes during the game sessions or, even better, creating an easy-to-use system to register these external events, especially when experiments are conducted in real classroom environments (as was the case in this study). This aspect should be reflected in the experimental design and should maintain the current pseudo-anonymization of the data (i.e., using anonymous codes instead of actual student identities). The collection of external data would allow us to determine whether the teacher helped a student during the session, or if the students interacted with each other. In this case, in order to minimize this type of interaction, players were not allowed to interact with their peers, and teachers were asked not to intervene unless the student asked for help and had stayed at the same level for more than 10 min. The collection of observational data also presents problems, as it is costly and complex for researchers, and is hardly scalable. We estimate that, for a typical class of 30 students (one per computer), it would take at least 3 people to record external events that may be interesting for the study; this number is influenced by the position of the computers and size of the room, as such experiments are generally conducted in uncontrolled actual environments.

The results also indicated some gender-related differences when it came to the interactions with the game and interest in continuing to play. Behavioral differences may be related to different learning approaches: boys seem to interact in a more exploratory way (making more interactions per minute) while girls seem to reflect more, taking more time to interact. These different behaviors have also been observed in other games, where boys tended to show a more active play mode [39]. This difference may be due to the affinity of players towards videogames and the number of hours they play daily as, as observed from the pre-test data, the use of videogames was more widespread among boys. Future work should focus on making the game more attractive to girls, in order to improve the results obtained in the survey on whether they would continue to play Articoding, with the ultimate aim of promoting the study of STEM among girls [40]. There is a need for studies that analyze the ways in which games are played and interacted with.

We believe that more research work is needed on GLA, providing assessments that other researchers can learn from. To date, emphasis has been placed on the sharing of analytical data; however, such sharing may face regulatory issues such as those associated with the European GDPR, as well as experimental designs. We believe that another way forward is to systematize the approach by attempting to utilize standards for trace capture (e.g., xAPI-SG) and to work on open-source tools—both for the tracker and the game itself—that other researchers can experiment with and learn from. It is also necessary to work on tools that help to visualize learning information during the use of games in class, particularly in a way that simple and useful for teachers.

Another aspect to be analyzed in future work is knowledge acquisition using the game and comparing it to other tools. For this evaluation, the game first needs to be in its final version, as any subsequent changes may invalidate previous evaluations. However, such an evaluation is not simple. On one hand, the game must be played in several sessions. It is not easy to find educational centers that are willing to allocate such an amount of time to the use of a tool before they are sure whether it is useful. Conducting these preliminary studies helps to gain the confidence of the schools. On the other hand, it is difficult to find validated instruments to study computational thinking and programming concepts. Due to these limitations, some studies use self-assessment or self-efficacy questionnaires without correlating the results to player behavior [45–47]. In future studies, an option would be to conduct a pre-post experiment with the tools that will be used in the new PISA25 test, and then conduct an analysis of how the game variables correlate with the results obtained through that test. Such studies would allow for the creation of tools that provide teachers with information about students' learning while using the game Articoding.

To conclude, although integrating GLA into a game and providing tools for the analysis and visualization of information is more expensive than simply using questionnaires or surveys, it brings great advantages in all phases of the game's lifecycle. Once the use of GLA is integrated into the game, it will be useful both in the evaluation and improvement phase of the game, as well as in its actual deployment in classes by teachers.

In particular, during the development and validation phases, the use of GLA allows us to detect design problems and gain a better understanding regarding how the game affects player behaviors. Meanwhile, during classroom use, it provides tools to teachers to better control their classes, identify students experiencing problems, and determine whether players are learning.

Author Contributions: Conceptualization, A.C.-M. and B.F.-M.; methodology, A.C.-M. and B.F.-M.; software, A.C.-M. and J.S.-B.; validation, A.C.-M., C.A.-F., and B.F.-M.; formal analysis, A.C.-M., C.A.-F., and J.S.-B.; investigation, A.C.-M. and B.F.-M.; resources, I.M.-O. and B.F.-M.; data curation, A.C.-M. and C.A.-F.; writing—original draft preparation, A.C.-M. and B.F.-M.; writing—review and editing, C.A.-F., J.S.-B., I.M.-O., B.F.-M., and A.C.-M.; visualization, A.C.-M.; supervision, I.M.-O. and B.F.-M.; project administration, I.M.-O. and B.F.-M.; funding acquisition, I.M.-O. and B.F.-M. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially funded by the Ministry of Education (PID2020-119620RB-I00; PID2023-149341OB-I00) and by the Telefonica-Complutense Honorary Chair on Digital Education and Serious Games.

Data Availability Statement: The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy and experimental design.

Acknowledgments: The authors would like to thank Dany Faouaz, Arturo García, and Álvaro Poyatos for programming the first version of Articoding.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Laamarti, F.; Eid, M.; El Saddik, A. An overview of serious games. *Int. J. Comput. Games Technol.* **2014**, *2014*, 358152. [CrossRef]
2. Connolly, T.M.; Boyle, E.A.; MacArthur, E.; Hainey, T.; Boyle, J.M. A systematic literature review of empirical evidence on computer games and serious games. *Comput. Educ.* **2012**, *59*, 661–686. [CrossRef]
3. Dörner, R.; Göbel, S.; Effelsberg, W.; Wiemeyer, J. *Serious Games*; Springer International Publishing: Cham, Switzerland, 2016. [CrossRef]
4. Wattanasoontorn, V.; Boada, I.; García, R.; Sbert, M. Serious games for health. *Entertain. Comput.* **2013**, *4*, 231–247. [CrossRef]
5. UNESCO. UNESCO Research Shows Women Career Scientists Still Face Gender Bias. Available online: https://en.unesco.org/news/unesco-research-shows-women-career-scientists-still-face-gender-bias (accessed on 14 March 2022).
6. Blumberg, S.; Krawina, M.; Mäkelä, E.; Soller, H. Women In Tech: The Best Bet to Solve Europe's Talent Shortage. McKinsey Digital. January 2023; pp. 1–10. Available online: https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/women-in-tech-the-best-bet-to-solve-europes-talent-shortage#/ (accessed on 1 August 2024).
7. Eurostat. More Women Join Science and Engineering Ranks. Available online: https://ec.europa.eu/eurostat/web/products-eurostat-news/-/edn-20220211-2 (accessed on 14 March 2022).
8. Schulte, C.; Sentance, S.; Sparmann, S.; Altin, R.; Friebroon-Yesharim, M.; Landman, M.; Rücker, M.T.; Satavlekar, S.; Siegel, A.; Tedre, M.; et al. Values and Beliefs Underpinning K-12 Computing Education. In Proceedings of the Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE, Milan, Italy, 8–10 July 2024; Association for Computing Machinery: New York, NY, USA, 2024; pp. 767–768. [CrossRef]
9. OECD. PISA 2025 Learning in the Digital World. Available online: https://www.oecd.org/en/topics/sub-issues/learning-in-the-digital-world/pisa-2025-learning-in-the-digital-world.html (accessed on 17 July 2024).
10. Shute, V.; Rahimi, S.; Smith, G. Game-Based Learning Analytics in Physics Playground. In *Data Analytics Approaches in Educational Games and Gamification Systems*; Springer: Singapore, 2019; pp. 69–93. [CrossRef]
11. Mojang, A.B.; TM Microsoft Corporation. Minecraft Education Edition. Available online: https://education.minecraft.net/ (accessed on 14 March 2022).

12. Electronic Arts. SimCityEDU: Pollution Challenge! Available online: https://www.ea.com/news/glasslab-launches-simcityedu?isLocalized=true (accessed on 9 June 2020).
13. Kahoot DragonBox. DragonBox. Available online: https://dragonbox.com/ (accessed on 14 March 2022).
14. Siew, N.M.; Geofrey, J.; Lee, B.N. Students' Algebraic Thinking and Attitudes Towards Algebra: The Effects of Game-Based Learning Using Dragonbox 12 + App. *Res. J. Math. Technol.* **2016**, *5*, 66–79.
15. Miljanovic, M.A.; Bradbury, J.S. A Review of Serious Games for Programming. In *Serious Games*; Göbel, S., Garcia-Agundez, A., Tregel, T., Ma, M., Hauge, J.B., Oliveira, M., Marsh, T., Caserman, P., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2018; Volume 11243, pp. 204–216. [CrossRef]
16. Hosein, A. Girls' video gaming behaviour and undergraduate degree selection: A secondary data analysis approach. *Comput. Human. Behav.* **2019**, *91*, 226–235. [CrossRef]
17. Freire, M.; Serrano-Laguna, Á.; Iglesias, B.M.; Martínez-Ortiz, I.; Moreno-Ger, P.; Fernández-Manjón, B. Game Learning Analytics: Learning Analytics for Serious Games. In *Learning, Design, and Technology*; Springer International Publishing: Cham, Switzerland, 2016; pp. 1–29. [CrossRef]
18. Elmoazen, R.; Saqr, M.; Khalil, M.; Wasson, B. Learning analytics in virtual laboratories: A systematic literature review of empirical research. *Smart Learn. Environ.* **2023**, *10*, 23. [CrossRef]
19. Kazimoglu, C. Enhancing Confidence in Using Computational Thinking Skills via Playing a Serious Game: A Case Study to Increase Motivation in Learning Computer Programming. *IEEE Access* **2020**, *8*, 221831–221851. [CrossRef]
20. Beltrán, E.J.G.; Arias, J.C.M. Mi Superpoder es la Programación: A tool for teaching programming to children and youth. *Sci. Comput. Program.* **2025**, *240*, 103198. [CrossRef]
21. Díaz, J.; López, J.A.; Sepúlveda, S.; Villegas, G.M.R.; Ahumada, D.; Moreira, F. Evaluating Aspects of Usability in Video Game-Based Programming Learning Platforms. *Procedia Comput. Sci.* **2021**, *181*, 247–254. [CrossRef]
22. Ouahbi, I.; Kaddari, F.; Darhmaoui, H.; Elachqar, A.; Lahmine, S. Learning Basic Programming Concepts by Creating Games with Scratch Programming Environment. *Procedia Soc. Behav. Sci.* **2015**, *191*, 1479–1482. [CrossRef]
23. Moreno-León, J.; Robles, G. Dr. Scratch: A Web Tool to Automatically Evaluate Scratch Projects. In Proceedings of the Workshop in Primary and Secondary Computing Education, London, UK, 9–11 November 2015; ACM: New York, NY, USA, 2015; pp. 132–133. [CrossRef]
24. Delozier, C.; Shey, J. Using Visual Programming Games to Study Novice Programmers. *Int. J. Serious Games* **2023**, *10*, 115–136. [CrossRef]
25. Gundersen, S.W.; Lampropoulos, G. Using Serious Games and Digital Games to Improve Students' Computational Thinking and Programming Skills in K-12 Education: A Systematic Literature Review. *Technologies* **2025**, *13*, 113. [CrossRef]
26. Tlili, A.; Chang, M. *Data Analytics Approaches in Educational Games and Gamification Systems: Summary, Challenges, and Future Insights*; Springer: Singapore, 2019. [CrossRef]
27. Liu, M.; Kang, J.; Liu, S.; Zou, W.; Hodson, J. Learning Analytics as an Assessment Tool in Serious Games: A Review of Literature. In *Serious Games and Edutainment Applications*; Springer International Publishing: Cham, Switzerland, 2017; pp. 537–563. [CrossRef]
28. Serrano-Laguna, Á.; Martínez-Ortiz, I.; Haag, J.; Regan, D.; Johnson, A.; Fernández-Manjón, B. Applying standards to systematize learning analytics in serious games. *Comput. Stand. Interfaces* **2017**, *50*, 116–123. [CrossRef]
29. Winslow, L.E. Programming pedagogy—A psychological overview. *ACM SIGCSE Bull.* **1996**, *28*, 17–22. [CrossRef]
30. Kwon, K. Novice programmer's misconception of programming reflected on problem-solving plans. *Int. J. Comput. Sci. Educ. Sch.* **2017**, *1*, 14–24. [CrossRef]
31. Robins, A.; Rountree, J.; Rountree, N. Learning and teaching programming: A review and discussion. *Int. J. Phytoremediat.* **2003**, *21*, 137–172. [CrossRef]
32. Shute, V.J.; Sun, C.; Asbell-Clarke, J. Demystifying computational thinking. *Educ. Res. Rev.* **2017**, *22*, 142–158. [CrossRef]
33. Alonso-Fernandez, C.; Perez-Colado, I.J.; Calvo-Morata, A.; Freire, M.; Ortiz, I.M.; Manjon, B.F. Applications of Simva to Simplify Serious Games Validation and Deployment. *IEEE Rev. Iberoam. Tecnol. Aprendiz.* **2020**, *15*, 161–170. [CrossRef]
34. Perez-Colado, I.; Pérez-Colado, V.M.; Martinez-Ortiz, I.; Freire, M.; Fernandez-Manjon, B. Simplifiying Serious Games Authoring and Validation with uAdventure and SIMVA. In Proceedings of the 2020 IEEE 20th International Conference on Advanced Learning Technologies (ICALT), Tartu, Estonia, 6–9 July 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 106–108. [CrossRef]
35. de Ruiter, L.E.; Bers, M.U. The Coding Stages Assessment: Development and validation of an instrument for assessing young children's proficiency in the ScratchJr programming language. *Comput. Sci. Educ.* **2022**, *32*, 388–417. [CrossRef]
36. Feigenspan, J.; Kastner, C.; Liebig, J.; Apel, S.; Hanenberg, S. Measuring programming experience. In Proceedings of the 2012 20th IEEE International Conference on Program Comprehension (ICPC), Passau, Germany, 11–13 June 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 73–82. [CrossRef]
37. Kao, G.Y.M.; Chiang, C.H.; Sun, C.T. Customizing scaffolds for game-based learning in physics: Impacts on knowledge acquisition and game design creativity. *Comput. Educ.* **2017**, *113*, 294–312. [CrossRef]

38. Alonso-Fernández, C.; Freire, M.; Martínez-Ortiz, I.; Fernández-Manjón, B. Improving evidence-based assessment of players using serious games. *Telemat. Inform.* **2021**, *60*, 101583. [CrossRef]
39. Rahimi, S.; Shute, V.; Kuba, R.; Dai, C.P.; Yang, X.; Smith, G.; Fernández, C.A. The use and effects of incentive systems on learning and performance in educational games. *Comput. Educ.* **2021**, *165*, 104135. [CrossRef]
40. Yusuf, A.; Noor, N.M.; Román-González, M. Interaction Patterns During Block-based Programming Activities Predict Computational Thinking: Analysis of the Differences in Gender, Cognitive Load, Spatial Ability, and Programming Proficiency. *AI Comput. Sci. Robot. Technol.* **2024**, *3*, 1–39. [CrossRef]
41. Elshiekh, R.; Butgerit, L. Using Gamification to Teach Students Programming Concepts. *Open Access Libr. J.* **2017**, *4*, 1–7. [CrossRef]
42. Giannakoulas, A.; Xinogalos, S. A pilot study on the effectiveness and acceptance of an educational game for teaching programming concepts to primary school students. *Educ. Inf. Technol.* **2018**, *23*, 2029–2052. [CrossRef]
43. Papadakis, S. Evaluating a game-development approach to teach introductory programming concepts in secondary education. *Int. J. Technol. Enhanc. Learn.* **2020**, *12*, 127. [CrossRef]
44. Mathrani, A.; Christian, S.; Ponder-Sutton, A. PlayIT: Game Based Learning Approach for Teaching Programming Concepts. *J. Educ. Technol. Soc.* **2016**, *19*, 5–17.
45. Holly, M.; Habich, L.; Seiser, M.; Glawogger, F.; Innerebner, K.; Kupsa, S.; Einwallner, P.; Pirker, J. FemQuest—An Interactive Multiplayer Game to Engage Girls in Programming. In Proceedings of the 2024 IEEE Conference on Games (CoG), Milan, Italy, 5–8 August 2024; pp. 1–8.
46. Hainey, T.; Baxter, G. A Serious game for programming in higher education. *Comput. Educ. X Real.* **2024**, *4*, 100061. [CrossRef]
47. Arnedo-Moreno, J.; García-Solórzano, D. Programming Fun(damentals): Using commercial video games to teach basic coding to adult learners. *Entertain. Comput.* **2025**, *52*, 100850. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.