Article

The Use of Educational Games in Programming Assignments: SQL Island as a Case Study

Stelios Xinogalos *◉ and Maya Satratzemi ◉

Department of Applied Informatics, University of Macedonia, GR-54636 Thessaloniki, Greece; maya@uom.edu.gr
* Correspondence: stelios@uom.edu.gr

Abstract: Educational games have attracted the interest of instructors and researchers in various fields of knowledge since they seem to motivate students, engage them in the educational process, and improve their performance. Computer programming is, without a doubt, cognitively demanding, and this has resulted in extended utilization of educational games. An important limitation of the relevant research lies in the fact that it is based on volunteer participants, while it is carried out as a side activity in programming courses and not in real-world course settings. In this article, a study investigating the utilization of the educational game SQL Island in the context of an assignment on the Structured Query Language (SQL) in a compulsory undergraduate course on Web programming is presented. Fifty-six students attending the course evaluated the player experience and perceived short-term learning through a questionnaire based on the MEEGA+ model for evaluating educational games targeted to computing education. The results both on player experience and perceived short-term learning were rather positive. Students' performance in the programming tasks of the game was very good, and the overall intervention proved to be effective. Moreover, conclusions were drawn on the constituent features of educational games that can be effectively used in the context of programming assignments.

Keywords: educational games; educational games for programming; SQL; programming assignments; player experience; perceived short-term learning

1. Introduction

Serious games, and more specifically educational games, are considered an important asset for the teaching and learning process nowadays. The use of games in the educational process increases students' motivation [1,2] and engagement [3], while it also enhances their performance [2].

The advantages of educational games are considered important for the teaching and learning of programming, which is a cognitively demanding field of knowledge causing various difficulties to students. This has resulted in a great number of serious or else educational games for various aspects of programming [4,5]. Some of these games are open source [6], and this gives the possibility of extending and/or adjusting them to the special needs of a specific class. However, as Silva and Silveira note, *"the amount of studies on educational games for teaching programming is low in the scientific literature"* [6] (p. 1). Moreover, in several studies, the educational context is not adequately described or is completely missing, or the participants are volunteers, and the study takes place as a side activity of a course. Consequently, there is a need for more studies that report experiences of utilizing games for teaching and learning programming in real classroom settings [7,8]. Specifically, research is required both in terms of the features that an educational game for programming should exhibit and the ways that it can be successfully integrated into the learning process.

The study presented in this article was designed to contribute to the limited research in the field [9–11] by investigating the outcomes of utilizing an educational game in the context of a compulsory undergraduate course on Web programming. Specifically, the

study aimed to investigate the outcomes of using an educational game, such as SQL Island [12] or SQL Mystery Murder [13], in the context of a programming assignment. SQL Island [12] was selected as a case study since it is freely available online; it includes several SQL statements utilized in the course and provides a personalized certificate if a user successfully completes the game. In order to contribute to the research on the challenges in designing and successfully integrating educational games in formal education, we studied: the perceived player experience and short-term learning using the MEEGA+ model [14]; students' performance in the game; features that are important for games utilized in programming assignments.

The rest of the article is organized as follows. Section 2 presents a review of research on educational games for teaching and learning SQL, as well as a brief comparison of freely available educational games for SQL programming. Section 3 summarizes important information for the methodology of the study: the context of the study; the participants; the research questions; the instruments used for data collection and analysis. Section 4 presents the results, which are discussed in Section 5. In Section 6, the main conclusions of the study are summarized, along with its limitations and plans for future research.

2. Related Work

2.1. Game-Based Learning of SQL

In this section, a literature review of studies on applying game-based learning (GBL) of SQL is presented. As Morales-Trujillo and García-Mireles [15] also note in their study, relevant work is limited. Actually, we were able to find just three studies that investigate the use of games in teaching and learning SQL [9–11] and two studies that investigate the use of gamification in web-based systems for practicing SQL [15,16]. Finally, a few studies on database design were also found. The main results of the studies on games and gamified systems of SQL are briefly presented.

Soflano et al. [9] investigated the learning effectiveness of three different versions of a 3D role-playing game (RPG) used in higher education for learning SQL. The three versions of the game consisted of: a non-adaptive version; a customized version based on students' learning styles as recorded in a questionnaire proposed by Felder-Silverman; a dynamically adaptive version of the game based on students' interactions during gameplay. The study included 120 higher education students and concluded that students that utilized the game for learning SQL, irrespectively of the version, achieved better learning outcomes in comparison with students that used a textbook for the same purpose. Moreover, students that used the adaptive version of the game managed to complete the assigned tasks in less time in comparison with the students that used the other two versions of the game.

In another study, Soflano et al. [10] studied the fluctuation of learning styles while playing a 3D role-playing game on SQL. This study utilized two distinct versions of the game: a non-adaptive version; and a dynamically adaptive version of the game based on students' interactions during gameplay. The study included 60 students and concluded that: the learning style recorded based on a learning style questionnaire is not always the same as the one recorded based on students' interactions in the game; in the case of learning style fluctuation, a strong correlation with the mistakes committed by the students was recorded.

Thanyaphongphat and Panjaburee [11] carried out a pilot study with 39 undergraduate students that used an adaptive online learning system that was designed so as to match students' learning styles and preferred digital media types. The SQL content was delivered to students through the online learning system utilizing three different types of digital media, namely Computer Assisted Instruction (CAI), Web-based Learning (WBL), and Digital Game-based Learning (DGBL). Out of the 39 students, 20 utilized DGBL for learning SQL. The study showed that students perceived the proposed system positively, found it easy to use, and believed that it could improve the learning process.

The aforementioned studies on using games for learning SQL were the only ones that we were able to find. The search for studies on utilizing gamification techniques as well for

teaching and learning SQL revealed just two more studies with undergraduate students attending database courses [15,16]. Morales-Trujillo and García-Mireles [15] carried out a quasi-experimental study with a group of 65 students that used a gamified version and a group of 74 students that used a non-gamified version of a web-based system for practicing SQL called QueryCompetition. The gamified version of QueryCompetition incorporated elements such as points and leaderboards. The results of the study showed a statistically significant improvement in performance for the group that used the gamified version of QueryCompetition, as well as higher motivation. Tahir et al. [16] studied the effects of adding game elements in the form of badges in an intelligent tutoring system called SQL-Tutor. The 77 students that participated in the study were divided into two groups, with one of them using the gamified and the other the non-gamified version. The results of the study showed no difference in terms of learning outcomes between the two groups of students.

Finally, there are also a few studies that focus on gamification techniques for issues connected to database courses in higher education, namely: learning database design [17,18]; assessing knowledge of database design [19]; learning the main features of NoSQL databases [20]. The main results of these studies are reviewed in Morales-Trujillo and García-Mireles [15].

It is clear that research on utilizing games in teaching and learning SQL is rather limited, and it is mainly focused on the effects of adaptation on motivation, user engagement, and learning outcomes. Although the research carried out so far is important, little is known about the utilization of games in the educational process under real-world situations. Moreover, the educational games utilized in existing studies are not freely available for usage in the classroom or replication of the studies. These facts provided the motivation for the study presented in this paper. In this study, we investigate the utilization of games in the context of programming assignments, which also has not been investigated to the best of our knowledge. The freely available online game SQL Island was utilized for a programming assignment in the context of a Web programming course as a case study. Further motivation for investigating the effects of using a game in an SQL assignment was provided by the fact that students faced difficulties in keeping concentrating on their courses during the COVID-19 pandemic lockdown. The game was anticipated to engage students and motivate them to participate more actively in the educational process.

2.2. Freely Available Games for SQL

A Google search for freely available games on SQL revealed just two games that are freely available through the web, namely SQL Island [12] and SQL Murder Mystery [13]. Moreover, the source code of SQL Murder Mystery is released under the MIT License and the educational content included in the project is released under Creative Commons CC BY-SA 4.0, as noted on the website of the project [13]. Both games have a scenario for keeping users motivated. In the case of SQL Murder Mystery, the player has to solve a crime mystery in SQL city, while in SQL Island, the player is informed that s/he has survived a plane crash and her/his mission is to make progress in the game in order to escape from the island. Both games incorporate a database that holds all the necessary information for reaching the game's goal, and the player writes SQL statements for retrieving the appropriate information from the database in response to specific challenges posed. In the case that an SQL statement is correct, the player obtains the necessary information for progressing in the game. In the case of providing a wrong SQL statement, a hint is provided in SQL Island, and the player retries, while in SQL Murder Mystery, the player has the ability to see the solution for several of the challenges. Both games provide educational material for SQL, but their type and extent differ significantly. SQL Murder Mystery actually provides a detailed tutorial for SQL interleaved with the game tasks, which constitutes it an appropriate resource for self-directed learning of SQL. On the other hand, SQL Island provides examples of SQL statements for the database holding all the information of the island along with their results prior to each task. Finally, SQL Island, in contrast with SQL

Murder Mystery, incorporates simple graphics for the characters (inhabitants of the island) that the player interacts with through the SQL statements s/he writes.

Taking into account the goals of the study, SQL Island seemed a good choice for an SQL assignment for the following reasons:

1. *Easy access through the web.* The only prerequisites for using the game are having a personal computer or laptop with minimum requirements and Internet access. Configuration and management of the environment is one of the challenges faced by students in relevant courses [21] that is not required.

2. *Simple Graphical User Interface (GUI).* SQL Island consists of the following areas presented in Figure 1: (a) the *graphical dialogue area* where the dialogue of the player with other inhabitants of the island is depicted; (b) the *results area* where the outcomes of SQL statements, either provided by the game as examples or written by the player in response to the challenges described, are presented; (c) the *editor area* where the player writes SQL statements; (d) the *database tables area* where the tables of the database are presented.

3. *Coverage of basic SQL commands, functions and operators.* The game covers the most fundamental SQL syntax that is used in an introductory SQL assignment: select, update, set, delete, where, is, and, or, null, group by, join, count, sum, avg.

4. *Appropriate examples* of SQL statements *and immediate feedback about the correctness of students' code* are presented.

5. *Self-paced practicing and assessment on SQL in a friendly environment.*

Figure 1. The GUI of SQL Island.

Apart from the advantages of SQL Island, the game has the following major limitations: the game is stateless, which means that the student has to start from the beginning if s/he does not manage to finish the game in one session; the game does not keep a record of students' code for inspection by the instructor. This is counterbalanced by the manageable number of programming tasks posed to the player and the fact that a certificate with the player's name is issued in the case of successfully completing all the tasks.

3. Methods

3.1. Context of the Study

The study took place in the context of a compulsory fourth semester "Web Programming" course at an Informatics Department where the duration of studies is four years. Specifically, the course was offered in the spring semester of the academic year 2019–2020. Due to the COVID-19 pandemic lockdown, both the lectures (2 h per week) and the lab (1 h per week) were taught online using Zoom.

The syllabus of the course includes the following server-side technologies: Web-server (Apache); server-side scripting (PHP); database backend (MySQL). The environment used in the labs was XAMPP (https://www.apachefriends.org/index.html (accessed on 4 June 2022)) since it integrates all the aforementioned technologies and, moreover, it is free and cross-platform.

Students were assigned three sets of exercises throughout the semester and a final project. The assignment that included the SQL Island game was announced after two labs (6th and 7th week) on SQL that utilized phpMyAdmin (https://www.phpmyadmin.net/ (accessed on 4 June 2022)) for practicing with SQL queries in ready-made databases distributed to students. Students had been introduced to SQL in a third-semester introductory course on Databases. The assignment was announced through the relevant tool of the Learning Management System utilized in the course, and students were asked to submit a document where they copied the SQL queries written for accomplishing the tasks in the game, as well as the certificate issued with their name when completing the game. The assignment also included another exercise that required using phpMyAdmin for writing and running SQL queries.

3.2. Research Questions

The aim of the study presented in this paper was twofold:

- Firstly, to investigate students' perceptions of the player experience and short-term learning in SQL Island (RQ1), as well as the possibility of using SQL Island in the context of a programming assignment on SQL (RQ2).
- Secondly, to draw some preliminary conclusions on features that a game used in the context of a programming assignment should have (RQ3).

In order to fulfill the first aim connected to the evaluation and the use of SQL Island in the context of an SQL assignment in a relevant undergraduate course, the following RQs were studied:

RQ1: *How do students evaluate the player experience and short-term learning in SQL Island?*
RQ2: *Can SQL Island be effectively used in the context of an SQL assignment?*
In order to fulfill the second aim, the following RQ was studied:
RQ3: *What features should a game utilized in the context of a programming assignment have?*

3.3. Participants

Fifty-six students attending the Web programming course submitted the assignment based on SQL Island and filled in an online questionnaire based on their experience with the game. The results of the questionnaire are analyzed in the next section. Thirty-nine participants (70%) were male, and the rest seventeen were female (30%). As mentioned, the course is taught in the spring semester of the second year. However, slightly more than half the students (55%) were in their second year of studies (aged 19–20), while more than one out of four students had surpassed the fourth year of studies, as shown in Figure 2. This is partly attributed to the difficulties that several students face with programming courses.
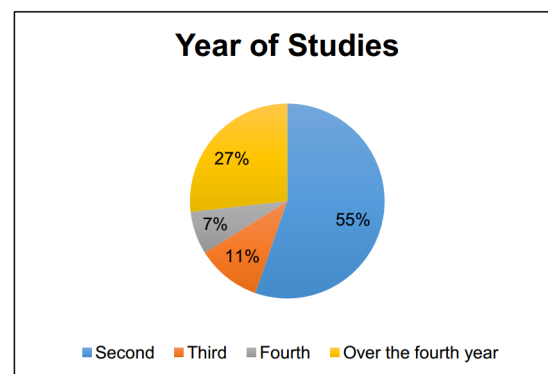
Figure 2. Distribution of students based on their year of studies.

3.4. Data Collection and Analysis

Two main sources of data were collected and analyzed to investigate the RQs of the study: students' performance in the assignments, including the one based on SQL Island; students' responses in an online questionnaire delivered as a Google form that was filled in after the experience with the SQL Island game.

The questionnaire was based on the MEEGA+ model that was proposed by Petri et al. [14] as an instrument for evaluating educational games. The reliability and validity (Cronbach's alpha $\alpha = 0.927$) of the model have been confirmed by data collected from 62 case studies with 1048 students, as well as the evaluation of 19 experts in educational games [22]. The MEEGA+ model has two main dimensions of evaluation, namely player experience and short-term learning. The player experience dimension is divided into several sub-dimensions. Based on this model, a questionnaire was proposed by Petri et al. [14] that was further amended and resulted in the questionnaire used in the presented study that consists of:

- Demographics questions (8 questions) including: sex; year of studies; non-digital games habits; digital-games habits; prior experience with serious/educational games; perceived knowledge of SQL (prior the assignment); statement of game completion; duration of gameplay.
- Player experience questions (27 questions). The player experience questions can be divided into the following dimensions: usability; confidence; challenge; satisfaction; fun; focused attention; relevance. Usability is further divided into: aesthetics; learnability; operability; accessibility; error prevention and recovery.
- Short-term learning questions (7 questions).
- Advantages and disadvantages of SQL Island and proposals for its utilization (3 open-type questions).

The questions on player experience and short-term learning were actually statements for which the participants were asked to express their degree of agreement on a 5-point Likert scale (1 = strongly disagree, 2 = disagree, 3 = indifferent, 4 = agree, 5 = strongly agree). For these questions, descriptive statistics were used (mean, standard deviation, and median), and the results were analyzed both quantitatively and qualitatively.

The open-type questions regarding the advantages and disadvantages of SQL Island and its potential educational usage aimed at investigating important issues both for the design and pedagogical usage of SQL Island, but even more importantly, serious games for programming. Students' responses to these questions were analyzed using elements of phenomenographic [23] and content analysis [24,25]. Representative keywords and phrases were highlighted using different colors in the Excel file that included the student responses collected from the Google form. This process was repeated several times until the main categories came to surface. Student responses were revisited and classified under the formulated categories that were based on students' own words. After making several iterations and considering the classification solid, frequencies and percentages were

calculated in order to have a clear picture of the main student trends/perceptions on the issues investigated.

4. Results

4.1. Student Game Play Habits and Prior SQL Knowledge

In the online questionnaire, nearly one out of four students stated that they play non-digital games on a weekly basis (18%) or even daily (5%), as shown in Figure 3a. When it comes to digital games (Figure 3b), more than half the students stated that they play such games on a weekly basis (30%) or daily (25%). Of course, this was not surprising since digital games are widely known to attract people of all ages. It is interesting, however, that although all the students stated that they play non-digital games, nearly one of ten students (9%) stated that they never play digital games and so it was possible that these students might not consider using a digital game in a course as a good idea.
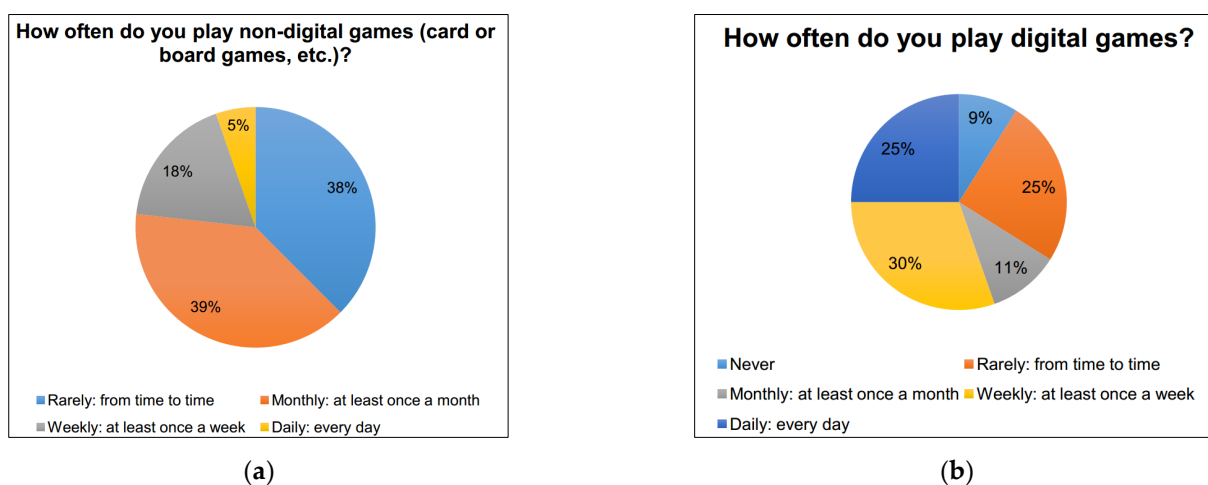

Figure 3. (**a**) Students' non-digital game habits. (**b**) Students' digital game habits.

Apart from students' experience with entertainment games, 66% ($N = 37$) of the students had also used a game in the context of another course or on their own for practicing in various fields of knowledge.

Finally, in a relevant 5-point Likert scale question, just 13% of the students stated that they had a good knowledge of SQL prior to playing the game, 46% of them had an average and 41% a poor knowledge (Mean = 2.7, Std.Dev. = 0.68, Median = 3).

4.2. Student Performance

According to students' responses in the questionnaire, 91% ($N = 51$) of them managed to finish the game, which means that they managed to write correctly all the SQL queries posed in the game. Of course, the students submitted the SQL statements they wrote for the queries, and they were corrected by the instructor, but this question was included as a means of checking the validity of the responses provided by the students to the questionnaire. Out of the five students that did not complete the game, two students correctly wrote 10 out of the 20 queries, and the other three students correctly wrote 15, 17, and 19 queries.

In Table 1, information about the exercises of each assignment carried out during the semester is presented. It is clear that the results of the assignments cannot be directly compared since they refer to different programming concepts that might pose different difficulties to students. Moreover, in all the exercises—except for the one based on SQL Island—students do not receive immediate feedback while writing their code. The increased number of students that excelled in the SQL Island exercise might be attributed to the fact that serious/educational games increase students' motivation [1,2] and engagement [3], but also to the fact that the game provides immediate feedback to students' attempts and

even hints, and this means that students are guided to correct their errors. The immediate feedback provided to students is considered important and can help them at the time of practicing and self-assessing their knowledge.

Table 1. Cumulative results for students' performance on the exercises (EX) of the course.

| | Assignment 1 | | | Assignment 2 | | Assignment 3 | |
	EX 1	EX 2	EX 3	EX 1	EX 2	EX 1	EX 2
Content	Basic structures, functions and arrays in PHP			Functions and forms in PHP	Text files in PHP	SQL queries	
Environment	XAMPP (or similar environment)			XAMPP		SQL Island	phpMyAdmin
Submissions	54	54	54	55	53	56	48
Completely correct	43	41	35	39	28	51	36
Percentage of correct submissions	80%	76%	65%	71%	53%	91%	75%

Studying the duration of playing SQL Island in the context of the corresponding exercise can also provide some interesting conclusions. The results of the relevant question included in the questionnaire are presented in Table 2. An interesting observation is that although some students managed to write all the SQL queries correctly and finish the game in 23 min, there were four students that devoted from 2 to 5 h in order to finish the game. This is definitely a clear indication of having difficulties with SQL, but also of engagement and persistence due to the game environment. Moreover, it is quite certain that without the immediate feedback provided by the game, these students would not have managed to reach a completely correct solution. This might explain the increased percentage of students that achieved the optimal solution in this exercise.

Table 2. Duration (in minutes) of playing SQL Island.

	Mean	Std.Dev.	Min	Max
All the participants ($N = 56$)	59	46	20	300
Participants that finished the game ($N = 51$)	56	44	23	300
Participants that did not finish the game ($N = 5$)	84	52	20	180

When studying the playing duration for the students that did not finish the game, we see that the average duration of game playing is 84 min, while for the students that finished the game, the average is 56 min. With the exception of one student that spent only 20 min in the game, the rest spent 70 (two students), 80 (one student), and 180 (one student) minutes. As was the case for the four students that finished the game after spending at least 2 h, the aforementioned four students (that spent more than one hour and despite this, they did not manage to finish the game) also faced difficulties with SQL. The time spent in the game is a clear indication of a long trial and error process. Taking into account that two students solved correctly half the queries and the other three were very close to completing the game, it is clear that even these students benefited from the game and managed to deal with their difficulties with SQL to a high degree through the immediate feedback and the hints provided by the game in real time.

4.3. Player Experience

In Table 3, learner perceptions on a five-point agreement scale for all the dimensions of player experience are summarized. As can be seen in Table 3, player experience was evaluated rather positively, with the exception of the focused attention dimension.

Table 3. Results regarding player experience.

Dimension	Subdimension	Item Description	Mean	Std.Dev.	Median
Usability	Aesthetics	The game design is attractive (interface. graphics. etc.).	3.5	0.83	4
		The text font and colors are well blended and consistent.	4.1	0.59	4
	Learnability	I needed to learn a few things before I could play the game.	3.3	1.19	4
		Learning to play this game was easy for me.	4.3	0.63	4
		I think that most people would learn to play this game very quickly.	3.7	0.97	4
	Operability	I think that the game is easy to play.	4.2	0.63	4
		The game rules are clear and easy to understand.	4.1	0.8	4
	Accessibility	The fonts (size and style) used in the game are easy to read.	4.4	0.65	4
		The colors used in the game are meaningful.	3.8	0.9	4
	Error prevention and recovery	The game prevents me from making mistakes.	3.6	0.94	4
		When I make a mistake it is easy to recover from it quickly.	3.7	0.92	4
Confidence		When I first looked at the game I had the impression that it would be easy for me.	3.6	0.82	4
		The contents and structure helped me to become confident that I would learn with this game.	4.2	0.59	4
Challenge		This game is appropriately challenging for me.	4.1	0.76	4
		The game provides new challenges (offers new obstacles. situations or variations) at an appropriate pace.	4.2	0.81	4
		The game does not become monotonous as it progresses (repetitive or boring tasks).	3.6	1	4
Satisfaction		Completing the game tasks gave me a satisfying feeling of accomplishment.	4.2	0.83	4
		It is due to my personal effort that I managed to advance in the game.	4.3	0.78	4
		I feel satisfied with the things that I learned from the game.	4.3	0.65	4
		I would recommend this game to my colleagues.	4.2	0.75	4
Fun		I had fun with the game.	4.2	0.71	4
		Something happened during the game (game elements. competition. etc.) which made me smile.	3.7	1.05	4
Focused Attention		There was something interesting at the beginning of the game that captured my attention.	3.6	0.91	4
		I was so involved in my gaming task that I lost track of time.	3.2	1.17	3
		I forgot about my immediate surroundings while playing this game.	2.6	0.98	3
Relevance		The game contents are relevant to my interests.	3.6	0.92	4
		It is clear to me how the contents of the game are related to the course.	4.6	0.56	5
		This game is an adequate teaching method for this course.	3.6	0.98	4
		I prefer learning with this game to learning through other ways (e.g., other teaching methods).	3.9	0.96	4

The participants were rather neutral to two out of the three aspects that constitute the focused attention dimension. Although the students agreed that *"there was something at the beginning of the game that captured their attention"* (median 4), they were divided as to whether they *"forgot about their surroundings"* and *"lost track of time"* while playing (median 3). However, we must note that achieving a high degree of immersion, although taken for granted in the case of entertaining games, is a difficult task for educational games. Analogous results were recorded for various programming games, even games with somewhat more professional graphics and richer gameplay mechanics than those in SQL Island, which were evaluated by end users utilizing the same questionnaire based on the MEEGA+ model. In the 2D game "Rise of the Java Emperor" [26], which is based on graphics provided by the popular game engine Unity, the user has to solve Java programming puzzles and battle with several enemies in order to defeat the Java Emperor and become his successor in the Java island. Although both the graphics and the gameplay mechanics in this game are more interesting in comparison with SQL Island, the thirty-three participants were indifferent as to whether they lost track of time and disagreed on forgetting about their immediate

surroundings while playing [26]. Exactly the same results were recorded in a study with the 2D platform game "jAVANT-GARDE" that aims at introducing novices to the main concepts of programming with Java [27]. In the 2D platform game "Py-rate Adventures", which focuses on basic programming concepts in Python, the participants disagreed both on "forgetting about their surroundings" and "losing track of time", although the player experience in terms of the rest of the dimensions was positive [28]. This implies that apart from the simplistic graphics and lack of rich gameplay mechanics, SQL Island poses some features that make it appealing to students. Students' replies to the open-type questions have revealed such successful features of SQL Island that will be analyzed in Section 4.5.

4.4. Perceived Short-Term Learning and Potential Usages of the Game

The results on students' perceived short-term learning were rather positive (median value of 4 for all statements), as shown in Table 4.

Table 4. Results on perceived short-term learning.

Item Description	Mean	Std.Dev.	Median
The game helped me comprehend new SQL statements.	4.3	0.53	4
The game helped me to comprehend better SQL statements that I already knew.	4.2	0.82	4
I learned something new through playing this game.	4	0.93	4
While playing I devoted time reading the extra information.	3.9	0.9	4
The game covered material that was important for the course.	4.2	0.83	4
The game provided a more efficient way of learning in comparison with other activities regarding this topic.	4	0.84	4
I would like such type of assignments in programming courses.	4.6	0.56	5

Based on the results on perceived short-term learning, nearly all the participants stated that they would like assignments based on games for programming courses (34% agree and 62.5% completely agree). However, the qualitative analysis of an open-type question regarding the circumstances in which SQL Island could be effectively used revealed other potential usages of SQL Island as well. Forty-one out of the fifty-six participants (73%) made proposals that were analyzed using content analysis, and the main categories of potential uses of the game are summarized in Table 5. For each one of the three main potential usages of SQL Island proposed by the participants, more specialized usages are also presented in descending order of frequency.

Table 5. Potential usages of SQL Island.

Using SQL Island for ...	Frequency	Percentage among the Participants that Made Proposals (N = 41)	Percentage among All the Participants (N = 56)
Learning SQL and Programming	30	73%	54%
• Learning SQL	17	41%	30%
• Introduction to SQL	9	22%	16%
• Learning programming	8	20%	14%
Practicing and consolidating knowledge	9	22%	16%
• Consolidating knowledge after studying	3	7.3%	5%
• Practicing existing knowledge	3	7.3%	5%
• Revision	3	7.3%	5%
Evaluation	4	10%	7%

4.5. Positive Aspects of the Game

In the open-ended question on the positive aspects of SQL Island, 35 out of the 56 participants (62.5%) responded. Their responses were analyzed qualitatively and qualitatively using content analysis, and the results are summarized in Table 6. More information about students' perceptions of the advantages of the game is presented in the paragraphs that follow using their own words translated from Greek.

Table 6. Positive aspects of SQL Island.

Positive Aspect	Frequency	Percentage among the Participants that Mentioned Positive Aspects (*N* = 35)	Percentage among All the Participants (*N* = 56)
Learning through the game	17	49%	30%
Interesting story	11	31%	20%
Entertaining	10	29%	18%
Easy to comprehend	9	26%	16%
Tips	7	20%	13%
Graphics and colors	5	14%	9%
Interactivity and immediacy	5	14%	9%
Satisfactory number of SQL statements	5	14%	9%
Appropriate for novices	3	9%	5%
Gradual increase of difficulty	3	9%	5%
Challenging	2	6%	4%
Evolves quickly	2	6%	4%

Learning through the game is considered to be one of the most positive advantages of SQL Island. The game *"helps in comprehending SQL in practice"* and also *"improving skills in SQL"* in the case of having prior knowledge. The game is *"more pleasant than reading a book or notes"* and *"more entertaining than reading slides"*. As one of the students stated, the game seems to promote *"learning in a creative and interactive way, has a satisfying level of difficulty that makes a student work and search more, an unprecedented way of learning that raises ones interest"*. We have to note, however, that this is actually the definition of serious games or, more specifically, educational games.

One of the constituent elements of an educational game is its scenario, as noted in established design frameworks for serious and educational games [3,29–32]. SQL Island has a simple scenario but yet *"it is an interesting game, with an interesting plot"*. The interesting story, in combination with the *"easy and comprehensible rules"*, *"the easy and comprehensible challenges and the use of appropriate examples"*, result in an entertaining game. When considering the educational aspects of the game, the tips provided during the challenges are an important feature and provide the necessary degree of scaffolding [33], according to nearly 20% of the students. The game *"provides background material each time that a challenge requires using a new SQL statement"*, as well as *"guidance in wrong excerpts of code"*.

The aforementioned aspects of SQL Island were the most frequent positive aspects mentioned by students. Other features that are actually proposed in serious games design frameworks [3,29–33] were also highlighted by students: *"pleasant graphics"* and *"nice colors that attract the interest and deter a potential attention distraction"*; *"the interactivity of the game creates an intrigue and persistence for achieving your goal and finishing the game"*; *"the difficulty of the challenges is gradual and this makes the game more accessible and leaves you space to adopt to its requirements for completing it"*. All these features make the game appropriate for novices.

4.6. Proposed Improvements of the Game

In the open-ended question on the potential improvements of SQL Island, 36 out of the 56 participants (64.2%) responded. Their responses were analyzed qualitatively and qualitatively using content analysis, and the results are summarized in Table 7 and

analyzed using excerpts from students' responses. The improvements proposed can be classified into two main categories: improvements in player experience; improvements in educational aspects of the game.

Table 7. Proposed improvements for SQL Island.

Improvement	Frequency	Percentage among the Participants that Proposed Improvements ($N = 36$)	Percentage among All the Participants ($N = 56$)
Better graphics	16	44%	29%
More SQL statements	11	31%	20%
Better error messages	7	19%	13%
More helpful hints	5	14%	9%
Theory—explanation of SQL statements	4	11%	7%
Sound effects and background music	3	8.3%	5%
Save point	3	8.3%	5%
Going back in the game	3	8.3%	5%

Regarding players' experience, the most referenced improvement refers to the game's graphics. Although students evaluated the aesthetics of the game (Table 3: median = 4) positively, it is clear that there is room for *"improvement of the graphics and enrichment of the environment of the game"*. For example, *"a more detailed animation, depicting the villages and better characters depiction would make the game more attractive"*. Furthermore, *"it would be interesting if there was background music for every correct or wrong answer"*, or, as another student commented, *"sound effects when the user answers correctly"*. However, as Michael and Chen [34] noted, the development of entertainment games differs from educational games in terms of resources, such as human resources, hardware, and multimedia material.

When it comes to improvements regarding the educational aspects of the game, the students mentioned that they would like the game to cover more SQL statements and also provide more scaffolding features. The game could have *"better error messages (debugging)"*, while *"if you make an error several times, the game could explain better the query"*. The hints could also be improved and, more specifically, *"the instructions could be clearer and in the case the player gets stuck there should be more helpful hints"*. *"Showing details for each statement that is currently being used in the game (at the side of the screen) in order to fully comprehend how each statement is used"* would also support students.

Although going back in a game might not be common in entertainment games, it seems that this feature is important for educational games. As one student stated, it is important *"to be able to choose at which one of the previous stages you want to go"*. *"A left arrow could be added in order to be able to see previous dialogues, so as to know what to do if we get stuck at some point"*. Finally, *"providing a save point"* in an educational game is important since there are students that face difficulties with the educational content and challenges, as noticed by the extended gameplay duration and the inability of some students to finish the game.

5. Discussion

In this section, the results recorded in the study are discussed in the context of the three research questions, and conclusions are drawn.

RQ1: How do students evaluate the player experience and short-term learning in SQL Island?

SQL Island is a simple game with basic graphics and limited game mechanics. However, based on its evaluation by 56 undergraduate students in the context of a compulsory SQL assignment, it turned out that it successfully fulfills its serious purpose, which is practicing SQL and self-assessing SQL knowledge.

When it comes to player experience, the majority of students evaluated positively (median = 4) all the relevant dimensions of the MEEGA+ model, with the exception of the dimension of focused attention. Specifically, students evaluated positively the

usability of SQL Island and all its constituent features, namely its aesthetics, learnability, operability, accessibility, and error prevention and recovery. Students were *confident* both about managing to finish the game and learning through playing, while the game was *challenging* enough to keep them engaged. Students felt *satisfied* with playing and learning within the game and would recommend it to colleagues. Although SQL Island is an educational game, students had *fun* with it, and, at the same time, its *relevance* to the course was clear to them. It is important that students consider the game as an adequate teaching method for the course and express their preference in learning with this game in comparison with other methods. In the relevant open-type question, the most referenced advantages of the game were the fact that students learn through the game that is based on an interesting story, is easy to comprehend (dialogues, tasks), and all these result in an entertaining experience.

As already mentioned, the feature that divided students was *focused attention* and specifically the immersion caused by the game. Maybe the use of better *graphics* and *animation*, which was one of the most referenced improvements for the game in the relevant open-type question, would improve this aspect of the game. *Background music* and *sound effects* could also have a positive impact.

The results of the evaluation were also positive regarding the *perceived short-term learning*. The students agreed that the game helped them comprehend new SQL elements or comprehend better SQL elements they already knew, and consequently, they learned something new through playing (median = 4). The game covered material that was important for the course, while students devoted time reading this material while playing (median = 4). Students agreed that the game provided a more efficient way of learning in comparison with other activities (median = 4), and they strongly agreed that they would like such type of activities in programming courses (median = 5). In the open-type question on the potential usages of SQL Island, students stated that the game could also be effectively used for learning SQL or even introducing students to SQL in a more intuitive and engaging way, practicing and consolidating knowledge, or even in an evaluation process. Practically, the game could be used in all aspects of the educational process.

In the open-type question regarding potential improvements for the game, students made proposals for several improvements that would enhance the learning aspects of the game. While some students (5 students) considered the number of SQL statements covered by the game as satisfactory, twice as many students (11 students) would like the game to cover *more SQL statements*. Students also mentioned various improvements that are connected with the *scaffolding* mechanism of the game: providing better error messages; more helpful hints, especially in the case of continuous inability to complete a task; providing brief and concise theory on the side of the game and explaining SQL statements; finally, having the ability to go back in the game for studying new material that was not absorbed or revisiting dialogues regarding the tasks that were not clearly understood.

RQ2: Can SQL Island be effectively used in the context of an SQL assignment?

Students' perceptions regarding the effectiveness of using SQL Island in the context of an SQL assignment were rather positive, as analyzed in the context of RQ1. Students were even more positive about using such types of assignments in programming courses.

Students' positive perceptions are also supported by the statistics of students' performance in this assignment and the assignments of the course collectively. Slightly more students submitted the assignment based on SQL Island in comparison with the other assignments, while 91% of them managed to submit a completely correct assignment. The percentage of completely correct submissions for the rest of the assignments ranged from 53% to 80%. Of course, this cannot be attributed solely to the type of the assignment but to the difficulty of the underlying programming concepts and the difficulty of the assignment per se as well. However, it is clear that the examples and the immediate feedback provided by the game for each SQL statement written by students, along with the tips/hints in the case of errors, support students in reaching the ultimate solution. The trial-and-error approach supported by the game, in combination with the story behind the game, keeps

students engaged and motivated. As shown by the qualitative analysis of students' responses to the assignment and the questionnaire, 4 out of the 51 students (8%) that finished the game spent more than two hours (mean playing time = 56 min), while 4 out of the 5 students (80%) that did not finish the game spent from 70 min to three hours. This is considered a clear indication of the engagement and motivation provided by both the entertaining and educational features of the game and the benefits it brings to students that face difficulties.

RQ3: What features should a game utilized in the context of a programming assignment have?
Based on the case study with SQL Island, we can conclude that the constituent elements of a game utilized in the context of a programming assignment are the following:

- The game should ideally be *Web-based* so as to be accessible from anywhere and at any time without the need for configuration.
- The game must have an interesting and intriguing *story* that keeps students engaged and motivated.
- Although high-quality professional graphics are not necessary, the game should have *appealing graphics and animated characters* to depict important elements of the story.
- Providing *well worked out examples of new programming constructs* prior to using them in the tasks helps students comprehend them.
- The *tasks*, or else challenges of the game, should be closely *connected to the story* of the game.
- The players should be able to *retry* in the case of failing a task and not be punished for their errors.
- Providing *hints* in the case of errors can help students become unstuck, face their difficulties with programming concepts, and keep on trying.
- The ability for *going back in the game* is necessary for giving players access to information connected to the educational content, the story and the tasks, as well as keeping them connected with the game.

Apart from the aforementioned features that are considered necessary for a game utilized in the context of a programming assignment, the following features will provide a better player experience and potentially better learning outcomes:

- The game should allow users to register in order to *save their progress* in the game. This is considered important for students that face difficulties with the underlying programming concepts and need more sessions for finishing the game or for games with many programming tasks.
- Saving students' progress in the game along with other data, such as time of playing, number of sessions, and erroneous attempts, would provide invaluable capabilities for instructors. These include *automatic assessment* of each individual student and presentation of *class statistics* utilizing the power of learning analytics.
- *Background music* can enhance the player experience.
- *Sound effects* that denote the correctness of the players' attempts in the tasks can help them keep informed about the outcomes of their efforts in the tasks of the game.
- *Providing brief and concise theory* at the side of the game can be helpful.
- Presenting details about the statement being executed at the side of the screen, an ability utilized successfully in programming microworlds [35] and known as *explanatory visualization* [36], can support students in comprehending the semantics of the underlying statements.

6. Conclusions

In this article, a study on utilizing the educational game SQL Island in an undergraduate course that uses elements of SQL was presented. In contrast with several controlled studies that evaluate various aspects of educational games (such as their effect on students' performance, motivation, and engagement) with volunteer students, the study presented was carried out under real-world circumstances, namely a compulsory undergraduate course on Web programming. Moreover, this study investigated the possibility of utilizing

an educational game in the context of a programming assignment. The ultimate goal was to support students in completing their tasks through the scaffolding and immediate feedback mechanisms of the game, as well as providing them with a more entertaining yet effective experience.

Students' performance (completion rate, playing time) in the game, as well as their perceptions regarding the player experience and short-term learning, were quantitatively and qualitatively analyzed. The results showed that students evaluated the player experience and short-term learning within the SQL Island game (RQ1) positively, although some improvements would be necessary in terms of focused attention or else immersion. Moreover, the utilization of SQL Island in one of the compulsory assignments of the course proved effective both in terms of students' achievements and expectations (RQ2). It is interesting to note that the programming assignment based on the game was both submitted and correctly implemented by more students in comparison with the rest of the assignments. What is considered even more important is that this study helped us draw conclusions on the features that are important for educational games targeted to the field of programming in order for them to be successfully integrated into the educational process (RQ3). Although SQL Island refers to SQL, we consider that its theme and overall rationale can be utilized in educational games for any programming language. Utilizing an interesting real-world scenario and having the student write code based on information provided by the game or with the aim of extracting information that is necessary for reaching specific goals within the game are key concepts in educational games for programming. In this sense, the outcomes of the study can be used as a roadmap for: utilizing SQL Island and other educational games in the context of programming assignments; extending the capabilities of SQL Island and designing new educational games for programming that can be effectively integrated into the educational process.

It is clear that further research is required with even more participants in order to confirm the findings of this study. We must note, however, that apart from the aforementioned limitation, the results can be considered trustworthy since the study took place in the context of a typical undergraduate programming course, and the fifty-six participants were regular attendants of this course. Moreover, both the game and the instruments of the study are freely available, and, consequently, the study can be easily replicated by interested instructors and researchers. Another interesting extension of this work would be to inform existing frameworks for designing educational games for programming [33] with specific features that fulfill the general requirements presented in their context.

References

1. 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]
2. Papastergiou, M. Digital game-based learning in high school computer science education: Impact on educational effectiveness and student motivation. *Comput. Educ.* **2009**, *52*, 1–12. [CrossRef]
3. de Freitas, S.; Jarvis, S. A Framework for Developing Serious Games to meet Learner Needs. In Proceedings of the Interservice/Industry Training, Simulation and Education Conference, Orlando, FL, USA, 4–7 December 2006.

4. Giannakoulas, A.; Xinogalos, S. A Review of Educational Games for Teaching Programming to Primary School Students. In *Handbook of Research on Tools for Teaching Computational Thinking in P-12 Education*; Kalogiannakis, M., Papadakis, S., Eds.; IGI Global: Hershey PA, USA, 2020; pp. 1–30. [CrossRef]

5. Miljanovic, M.A.; Bradbury, J.S. A review of serious games for programming. In *Joint International Conference on Serious Games*; Springer: Cham, Switzerland, 2018; pp. 204–216.

6. Silva, J.; Silveira, I. A Systematic Review on Open Educational Games for Programming Learning and Teaching. *Int. J. Emerg. Technol. Learn. (IJET)* **2020**, *15*, 156–172. [CrossRef]

7. Jakoš, F.; Verber, D. Learning basic programing skills with educational games: A case of primary schools in Slovenia. *J. Educ. Comput. Res.* **2017**, *55*, 673–698. [CrossRef]

8. Malliarakis, C.; Satratzemi, M.; Xinogalos, S. CMX: The Effects of an Educational MMORPG on Learning and Teaching Computer Programming. *IEEE Trans. Learn. Technol.* **2017**, *10*, 219–235. [CrossRef]

9. Soflano, M.; Connolly, T.M.; Hainey, T. An application of adaptive games-based learning based on learning style to teach SQL. *Comput. Educ.* **2015**, *86*, 192–211. [CrossRef]

10. Soflano, M.; Connolly, T.M.; Hainey, T. Learning style analysis in adaptive GBL application to teach SQL. *Comput. Educ.* **2015**, *86*, 105–119. [CrossRef]

11. Thanyaphongphat, J.; Panjaburee, P. Effects of online learning with matching learning styles and preferred digital media types on university students' perceptions. In Proceedings of the 2017 6th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI), Hamamatsu, Japan, 9–13 July 2017; pp. 552–555.

12. SQL Island. Game was Developed by Johannes Schildgen at the University of Technology, Kaiserslautern in the Heterogenious Information System Group. Available online: https://sql-island.informatik.uni-kl.de/ (accessed on 4 June 2022).

13. Knight Lab. SQL Murder Mystery. *The Northwestern University.* Available online: https://mystery.knightlab.com/ (accessed on 4 June 2022).

14. Petri, G.; von Wangenheim, C.; Borgatto, A. MEEGA+: An Evolution of a Model for the Evaluation of Educational Games. Brazilian Institute for Digital Convergence. *Tech. Rep.* **2016**, *3*, 1–40.

15. Morales-Trujillo, M.E.; García-Mireles, G.A. Gamification and SQL: An empirical study on student performance in a database course. *ACM Trans. Comput. Educ. (TOCE)* **2020**, *21*, 1–29. [CrossRef]

16. Tahir, F.; Mitrovic, A.; Sotardi, V. Investigating the Effects of Gamifying SQL-Tutor. In Proceedings of the Virtual Conference: 28th International Conference on Computers in Education, Virtual Conference, 23–27 November 2020; pp. 416–425.

17. Abdool, A.; Ringis, D.; Maharajh, A.; Sirju, L.; Abdool, H. DataRPG: Improving student motivation in data science through gaming elements. In Proceedings of the 2017 IEEE Frontiers in Education Conference (FIE), Indianapolis, IN, USA, 18–21 October 2017; pp. 1–5.

18. Santhanam, R.; Liu, D.; Shen, W.C.M. Research Note—Gamification of technology-mediated training: Not all competitions are the same. *Inf. Syst. Res.* **2016**, *27*, 453–465. [CrossRef]

19. Rahman, R.A.; Ahmad, S.; Hashim, U.R. The effectiveness of gamification technique for higher education students engagement in polytechnic Muadzam Shah Pahang, Malaysia. *Int. J. Educ. Technol. High Educ.* **2018**, *15*, 1–16. [CrossRef]

20. Butgereit, L. Four NoSQLs in four fun fortnights: Exploring NoSQLs in a corporate IT environment. In Proceedings of the Annual Conference of the South African Institute of Computer Scientists and Information Technologists, Johannesburg, South Africa, 26–28 September 2016; pp. 1–6.

21. Xinogalos, S.; Kaskalis, T. The Challenges of Teaching Web Programming: Literature Review and Proposed Guidelines. In Proceedings of the 8th International Conference on Web Information Systems and Technologies (WEBIST 2012), Porto, Portugal, 18–21 April 2012; pp. 207–212.

22. Petri, G.; von Wangenheim, C.G. MEEGA+: A Method for the Evaluation of the Quality of Games for Computing Education. In Proceedings of the SBGames, Rio de Janeiro, Brazil, 28–31 October 2019; pp. 28–31.

23. Marton, F. Phenomenography—describing conceptions of the world around us. *Instr. Sci.* **1981**, *10*, 177–200. [CrossRef]

24. Stemler, S. An overview of content analysis. *Pract. Assess. Res. Eval.* **2000**, *7*, 17.

25. Weber, R.P. *Basic Content Analysis*; Sage: Newbury Park, CA, USA; London, UK; New Delhi, India, 1990.

26. Tsiotras, D.; Xinogalos, S. Investigating the Perceived Player Experience and Short-term Learning of the Text-based Java Programming Serious Game "Rise of the Java Emperor". *Inform. Educ.* **2021**, *20*, 153–170. [CrossRef]

27. Galgouranas, S.; Xinogalos, S. jAVANT-GARDE: A Cross-Platform Serious Game for an Introduction to Programming with Java. *Simul. Gaming* **2018**, *49*, 751–767. [CrossRef]

28. Sideris, G.; Xinogalos, S. PY-RATE ADVENTURES: A 2D Platform Serious Game for Learning the Basic Concepts of Programming with Python. *Simul. Gaming* **2019**, *50*, 754–770. [CrossRef]

29. Kiili, K. Content creation challenges and flow experience in educational games: The IT-Emperor case. *Internet High. Educ.* **2005**, *8*, 183–198. [CrossRef]

30. Salen, K.; Zimmerman, E. *Rules of Play: Game Design Fundamentals*; The MIT Press: Cambridge, MA, USA, 2004.

31. Song, M.; Zhang, S. EFM: A model for educational game design. In Proceedings of the International Conference on Technologies for e-Learning and Digital Entertainment, Nanjing, China, 25–27 June 2008; pp. 509–517.

32. Yusoff, A.; Crowder, R.; Gilbert, L.; Wills, G. A conceptual framework for serious games. In Proceedings of the 2009 Ninth IEEE International Conference on Advanced Learning Technologies, Riga, Latvia, 15–17 July 2009; pp. 21–23.

33. Malliarakis, C.; Satratzemi, M.; Xinogalos, S. Designing educational games for computer programming: A holistic framework. *Electron. J. e-Learn.* **2014**, *12*, 281–297.

34. Michael, D.R.; Chen, S.L. *Serious Games: Games That Educate, Train, and Inform*; Thomson Course Technology: Boston, MA, USA, 2006.

35. Xinogalos, S.; Satratzemi, M.; Dagdilelis, V. An Introduction to object-oriented programming with a didactic microworld: ObjectKarel. *Comput. Educ.* **2006**, *47*, 148–171. [CrossRef]

36. Brusilovsky, P. Program visualization as a debugging tool for novices. In Proceedings of the ACM INTERACT93 and CHI93 Conference Companion on Human Factors in Computing Systems, Amsterdam, The Netherlands, 24–29 April 1993; pp. 29–30.