applied sciences

MDPI

Article

Model for Semi-Automatic Serious Games Generation

Pedro Omar Silva-Vásquez [1], Viviana Yarel Rosales-Morales [2,*], Edgard Benítez-Guerrero [1], Giner Alor-Hernández [3], Carmen Mezura-Godoy [1] and Luis Gerardo Montané-Jiménez [1]

[1] Facultad de Estadística e Informática, Universidad Veracruzana, Av. Xalapa, Col. Obrero Campesina, Xalapa-Enríquez 91020, Veracruz, Mexico
[2] Facultad de Estadística e Informática, CONACYT-Universidad Veracruzana, Av. Xalapa, Col. Obrero Campesina, Xalapa-Enríquez 91020, Veracruz, Mexico
[3] Technological Institute of Orizaba, Tecnológico Nacional de México, Av. Oriente 9, 852, Col. Emiliano Zapata, Orizaba 94320, Veracruz, Mexico
* Correspondence: vivrosales@uv.mx

Abstract: Serious games (SG), (video games with an educational purpose), provide teachers with tools to strengthen their students' knowledge. Developing a SG requires knowledge, time, and effort. As a result, specialized tools to aid in the development process are needed. This work presents a model for the development of SG in the platformer genre. A tool implementing the model is introduced as a proof of concept. A SG was generated using this tool, which in turn was evaluated in terms of gameplay, mechanics, story, and usability. The evaluation results show that the SG has the minimum elements requested by an audience of students, who were expecting a game with both entertaining and educational value. Furthermore, the results are satisfactory in three out of four areas, showing that there are opportunities for improvement regarding the game's story. Our work intends to improve the development times of new SG, as well as to make them easier to develop by both software engineers and teachers who wish to implement them in their classrooms.

Keywords: semi-automatic code generation; serious games; model-driven game development; platform game

Citation: Silva-Vásquez, P.O.; Rosales-Morales, V.Y.; Benítez-Guerrero, E.; Alor-Hernández G.; Mezura-Godoy, C.; Montané-Jiménez, L.G. Model for Semi-Automatic Serious Games Generation. *Appl. Sci.* **2023**, *13*, 5158. https://doi.org/10.3390/app13085158

Academic Editor: Andrea Prati

Received: 20 December 2022
Revised: 28 January 2023
Accepted: 3 February 2023
Published: 20 April 2023

1. Introduction

Serious games (SGs), also known as educational games or learning games, are video games aimed at acquiring new knowledge or training skills, beyond their entertainment value [1]. They serve as alternative tools to transmit knowledge to people [2]. SGs blend pedagogical principles with the engaging features usually present in video games, such as game mechanics, in order to motivate and guide the user in learning tasks [1]. SGs in school classrooms have proliferated over the past decade, with many proven benefits [3], They have been used to cover a wide range of topics, such as science, health, history, and business practices [1].

"Duolingo" is a popular example of a SG: a language learning application that uses a system of rewards and challenges to motivate users to continue practicing and improving their skills. Another example is "Kahoot!", an educational platform that allows teachers to create and use interactive games in the classroom to teach a variety of subjects. Or even "Minecraft: Education Edition"; a spin-off of the popular video game "Minecraft", specifically designed to allow students to learn science, math, history, and literature.

SG have proven to be useful tools to reinforce students' learning in an interactive and motivational way. Through the combination of game elements and learning mechanics, SGs can help students to retain and apply information more effectively. However, according to [4–8], their development is expensive. The production of SGs requires a high level of planning, communication, and organization between multidisciplinary teams, in an effort to avoid costly delays and failures [8]. In general, existing models and methodologies to develop SGs ignore the multidisciplinary nature of the task, requiring the participation

of designers, artists, programmers, and testers to mention a few. It is also necessary to consider educators, who serve as domain experts and have the skills to synthesize and manage information intended for prospective students. Yet, expert participation is often ignored [3]. On the other hand, as the development of SGs requires extensive technical knowledge, it is difficult for teachers with little experience in video games to create them [9]. Teachers needing a SG are forced to outsource game development, which requires time and effort. A possible solution is to provide teachers with models and tools that capture pedagogical and technical knowledge to make the development process more accessible.

In this paper, we present a model that organizes the design of an educationally oriented platform game. This, through a set of design entities that represent the features most frequently considered in the production of engaging educational game experiences. The organization of the elements of the model facilitates reusing the pieces of the obtained designs, to quickly produce variants. In addition, we implemented a tool that works as a proof of concept for the proposed model. This tool takes the form of a wizard, where the user defines a high-level configuration for a proposed SG, and the tool generates its code automatically. The product is delivered to the user as a package that must be entered into the game development engine GDevelop, to finish the export process and obtain an executable with the complete game.

As a use case we considered the creation of a game on the topic of relational algebra for database systems, a topic reported as difficult to learn by our users. The resulting game provides a story (an adventure) through its platforming mechanics, comprising a main character, enemies, and obstacles. Throughout the adventure, the main character has to answer questions regarding the subject of relational algebra to continue advancing in his journey. The game consists of several levels of increasing difficulty, reflecting the student's progress in the subject matter. Variants of this SG can be created through the tool. Our contribution is placed in the field of automatic code generation for SGs, which indirectly contributes to game-based learning.

The rest of this paper is organized as follows. Section 2 discusses related works. Section 3 presents our model proposal with a detailed description of its processes and components. Section 4 explains the use case, while Section 5 shows a more detailed analysis of the results. Finally, Section 6 concludes this paper.

2. Related Work

In the following, previous work is presented to provide a context for the reader concerning our research and proposal. This section is divided into two parts. The first section presents the existing models for the development of SG, and the second part presents the tools that support the development of SG.

2.1. Models for Developing Serious Games

Marchiori et al. [10] presented the DSVL model that simplifies the development of SG for educators who do not have programming skills. It draws on concepts of visual language and narrative theory to create a description of games that is easy to understand and maintain. The DSVL model is limited to adventure games. This model facilitates rapid prototyping to enable early evaluation by customers and users. Despite the efforts to emphasize the narrative processes of the video game, in general, there is a lack of work on the pedagogical part. Marne et al. [11] built a framework composed of six facets of SG design: pedagogical objectives, simulation domain, simulation interactions, problems and progression, decorum, and conditions of use. The final result is an adaptation to the needs of the teachers, allowing them to understand the objectives, means, and methods of game experts. The work presented above focuses on a single type of video game to integrate a unique narrative that is immersive for the user with a dynamic that can change according to the main topic of the SG. Compared to the last proposal of Marne et al. [11], our model depends on three phases.

Refs. [12–14] reported models that are concerned with both the pedagogical and the entertainment part of the generation of a SG.

Ref. [12] presents a conceptual model called activity theory based SGs model (ATMSG), which describes the elements of the video game, helps to identify and understand the functions of each component of the game and the educational objectives of the SG. In [13], the authors focused on a problem that educational game designers have to deal with: the technical complexity of the development. Ref. [13] proposes GREM (game rules scenario model), a game model that uses the features that are most frequently found in the literature. The elements of this model are arranged in two different and independent sub-models: the game rules model and the scenario model.

The learning mechanics–game mechanics (LM-GM) model proposed in [14] includes predefined game mechanics and pedagogical elements that can also be useful for teachers to evaluate the effectiveness of a given game and better understand how to apply it in educational settings. Results with users demonstrate the advantages of this approach.

Some models found are targeted to specific users. In [15], the authors present a model for the design of kind of SGs for hearing-impaired children. The results obtained in the user experience evaluation helped to identify aspects of the game mechanics. The results produced high scores on the questionnaires, indicating success.

For example, in [16], the authors focus on persuasive strategies. Video games were adapted to the personality type of the players. Demonstrating that it improved the effectiveness of the games. Other models were found: pedagogical and entertainment. The authors of [3] presented a model for the development of educational games based on six phases: design of chapters, design of scenes (scenarios, characters, actions, and dialogues), design of educational challenges in the game, design of adaptation, design of emotional experience and design of collaboration.

Other models focus on the immersive part for SG. Ref. [17] proposes a model for SGs, called FRACH, that contemplates a structure of inputs and outputs for interaction in SG. In the end, the game is effective for knowledge acquisition.

In [18], on the choice of game type, a model for developing a video game is proposed. The authors focused on game modes, actions, challenges, goals, rewards/penalties, and a story. The game elements–attributes model (GEAM) proved to be a promising framework.

In the research work of [19], the authors proposed iPlus, a SG design methodology based on a participatory, flexible, and user-centered approach. Not all SG that have been developed have applied appropriate design methodologies that incorporate both the entertainment mechanics and the serious component.

The authors of [20] conducted an analysis of existing models, focusing on model-based software development. They present a latent Dirichlet allocation (LDA) approach for feature localization in software models, models for code generation, and interpreted models for a commercial video game. The analysis helped to better understand and characterize these models.

Our proposed model contemplates aspects such as clearly stating the objectives of the game, the challenges, the rewards, and the story that can involve the end user, facilitating the work of designing a video game to teachers or software developers.

Other works propose shortening the development time by giving a list of requirements, such as [21], where the authors report a model-generation approach (EMoGen). Using this approach requires only five hours compared to ten months of developer work. In [22], the authors present a game tool design that supports learning. It was concluded that the tool did not contribute significantly to students' learning performance, as it is only a support and reinforcement of knowledge.

Compared to previous works, our proposal has only three phases, while others have six. The rules of the game are implicit in the mechanics of the game, which is a platform game, so the scenario is intended for platform interaction. In our model, we consider a SG as a reinforcement tool, not as a learning tool. The teacher had to teach the subject and direct the students to reinforce the knowledge acquired in the classroom with the help of

the SG. In our model, we focused on the widespread use of a video game that many users were familiar with, so we chose a platform-type mechanic in the style of popular games, such as Mario Bros, Cup head, Celeste, or Rayman. In our model, the focus is on the choice of the main character, the story, the selection of the scenarios, and the uploading of the questions to the question bank.

For the model presented in this research, we give a complete video game structure, where the teacher only has to share the educational reinforcement material to be implemented in the platform video game, cutting the development process in the video games, because in the end, he would have a code that can be supervised in a video game engine, such as Gdevelop, which is presented as a friendly and easy to learn interface.

2.2. Tools for Developing Serious Games

In the search for tools to help developing a SG, we found the proposal of [23], which presents StoryTec, a digital storytelling platform for the creation and experimentation of non-linear interactive stories. The platform focuses on two specific parts: the story editor and the standardized descriptive format for an interactive story. Favorable results were obtained, but there is still work to be done on usability, stability, and scalability. Another proposal is that of [24], where the authors present a system for the creation of platform game levels. It integrates the concepts that can be found in this type of game. They employed some techniques that allow the automatic generation of levels. The system was evaluated with satisfactory results.

In [25], the authors present a level editing tool that allows the human design of levels and testing of automatic generation algorithms. An adapted version of the editor was implemented for semi-automatic level creation, where the designer can simply define the type of content he/she wants in the form of quests and missions and the system creates the corresponding level structure. The tools in [24,25] sought to solve the problem of SG development. In [26], the authors present modding for games as a pedagogical practice in a game design course. In particular, this approach is beneficial, as it allows students to circumvent technological barriers. With two different mods of the same platform game, the authors can allow students to engage in video game design to explore the relationship between mechanics and meaningful play.

The authors of [7] present an authoring tool for developing game designs that can be exported to XML files, and a game engine capable of interpreting such files. This facilitates the work of designers. In evaluations, its feasibility and acceptability by both technical and non-technical users were validated. The authors of [27] presented uAdventure: a SG editor built on top of the Unity game engine that enables the creation of educational adventure games without programming. uAdventure improves the SG lifecycle by reducing authoring and maintenance costs, as it evolves with the unity game engine. Ref. [28] presents a graphical editor that provides high-level models representing the gamification strategy, its deployment, and monitoring. These models contain the definitions of event patterns that are automatically transformed into code. The proposal can be used in learning management systems (LMS), such as Moodle. The framework presented in [29] provided a toolbox to (i) create 2D platform levels, (ii) estimate the difficulty and success probability of single jump actions, and (iii) to evaluate the difficulty using a set of metrics. The results were obtained from developers and players who approved the framework. Educators need software platforms for the automated construction and flexible customization of such games. Ref. [30] presents a platform called Maze Builder, based on Unity 3D, which automatically and easily generates video games of mazes. The results are very positive and encouraging in terms of the use of the Maze Builder platform by specialists who are not computer science professionals. The authors of [31] describe their smart adaptive video games for education (APOGEE) platform for the automated construction of educational video games. The construction process of building process includes three stages: game design, game validation, and game generation. The tool monitors platform data and

processes, which will make it easier for platform users to create more adaptable, effective, and efficient video maze games for education.

The tool presented in [32] consist of two components: (i) an interface that allows the user to design the game and capture the motion data, and (ii) a customizable game for learning and training using commercially available motion capture sensors, such as Microsoft Kinect. The game is automatically configured based on the output of the game design interface. The results showed that the use of a game-like application could be efficient, as positive feedback was obtained. Modding is a form of production in which players experiment by developing and conceptualizing the modification of a video game. The study [33] presents modifications to video games. The results show informal learning obtained by the participant's performance and skill performance, teamwork, or problem solving. The research in [34] presents a level-generation system, which uses a graph structure, the automatic detection of level structures, and graph grammars. Experimental analysis shows that the proposed system can shorten the development and design times of a platform game. The authors of [9] present the authentic role-playing-game quest system (ARQS), a tool to support the implementation of a serious role-playing game (RPG). It was very well accepted in the conducted test. The paper [35] proposes a new process for developing augmented reality SGs (ARSGs), which comprises three phases: analysis, configuration, and generation. It automatically generates the application using augmented reality with the educational elements entered by the teacher. An evaluation was performed with teachers and developers, and the results were positive.

The works previously presented consisted of tools for the automatic or semi-automatic generation of video games. The most popular ones were those that present graphic novels. We selected another type of game: platform games. In this research, we aim to address the challenge of software developers and teachers facing difficulties in designing levels, missions, or challenges for SGs. Our tool proposal seeks to streamline the development process by providing a pre-structured game template with a bank of questions, eliminating the need for expert knowledge in video game design. We chose a popular and widely recognized platform game as the base for our tool, in order to make it easily accessible for users who may not be familiar with the subject of video games. Additionally, we propose using XML formats for the structure of the code in order to make the process semi-automatic and easily interpretable by the GDevelop game engine.

Our goal is to create a tool that is user-friendly and accessible to those who may not have prior experience in the field of SG development. We aim to limit the design aspects that may present a learning curve for the end user in order to make the tool as easy to use as possible.

3. Model Proposal

In building the model, we considered related works of the state of the art. We decided to start with a platform game, and it was divided into essential and objective parts that represent it: the main character, enemies, obstacles, rewards, and story. The graphics must have a 2D perspective. In terms of mechanics, it was assumed that the player has a set number of lives or a certain percentage of damage. If the user is hit by an enemy or an obstacle, they will lose a life or have their life percentage reduced. The main character only has three movements: left, right, and jump. Each level has an objective to be achieved. Additionally, an educational interaction was included, where information was presented on the main theme of the game and a multiple-choice question was asked of the user. Incorrect answers reduce the player's life.

For the SG configuration, the user options were shortened, focusing on minimal modifications and the educational theme that the teacher wants to teach. The model has a basic design to guide the user and follow the process steps. Each phase has a specific function, and a flowchart was followed to generate the code for the serious platform game to be loaded in the GDevelop game engine.

3.1. Phase A: Serious Game Configuration

This phase is made up of three sub-phases. The first one is "customization", where the features that a user wishes for his SG are captured: the name of the video game, main character, story, number of levels, and scenarios for each level. The second sub-phase is "validation", the where requirements for the SG are checked and the configuration selected by the user is confirmed. The third phase is "creation and saving", where an XML file that will serve as a backup of the user's choices regarding their preferences for the SG is generated.

In Figure 1, which deals with the configuration of the SG, consists of 11 steps:

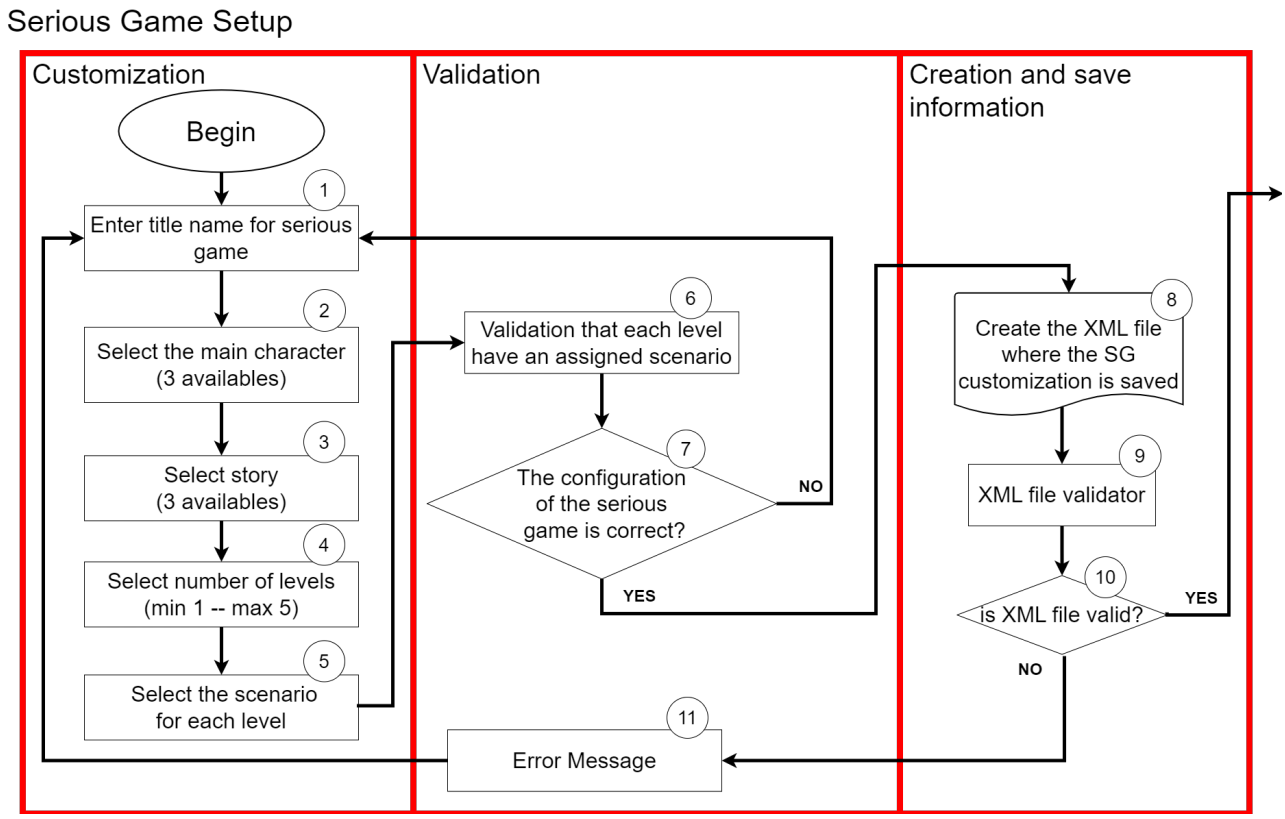

Figure 1. The first part of the flowchart is where SG setup occurs.

1. Enter the name of the video game that will appear as the title of the SG.
2. Select the main character. You will be able to choose between three different characters that can be the main character of the story. There are three main characters available: Ximena, Ruben and Nino.
3. Select the story. The user will be able to select among the three different stories available for the SG:
   - A quest storyline.
   - A rescue storyline.
   - An invasion storyline.
4. Select the number of levels. The user must select how many levels the student will be able to interact with, one level as the minimum and five as the maximum. The user must consider that at least each level must have three questions that in the next module must be attached to each of the selected levels.
5. In the next step, the five available scenarios for the game are presented. One scenario must be associated with a game level:
   - Forest.
   - Desert.

- Snowy mountain.
- Cemetery.
- Pantheon.

6. Validate that each level that the video game wants has a scenario assigned to it.
7. Confirm if the user wants this configuration. If it is true, it advances to step number 8; if it is false, it goes back to the configuration from step number 1.
8. An XML file is created with the tags that will be used for the generation of the code and the information of the preferences selected by the user for the serious video game he/she wants is saved.
9. The validation that the XML file was created and that the corresponding information was saved is made.
10. A validation of the XML file is performed, which determines if there are any errors in the XML file. If everything is saved, it will go to step number 12; otherwise, it will go to step number 11.
11. An error message is presented to inform the user what went wrong.

3.2. Phase B Content Configuration

In this phase, the user adds the educational content that he wishes to appear throughout the video game with the questions with which the user interacts and the information that functions as support for the educational part. The phase consists of four sub-phases: The first phase is divided into two options for the user to add the content for the SG using a CSV file, either automatically or manually via a form.The third phase is the "validation phase." This operation consists of verifying that the CSV file meets the functional requirements. This same validation is performed for the manual option of the form; in the case of noncompliance with the requirements, the non-compliance must be solved to make it functional. The fourth phase is the "saving" phase, where the questions are saved in a CSV and the XML file is updated. It is validated that it is saved in order to proceed to the next phase, as shown in Figure 2.

In the previous phase, the XML file was created, and the educational content will be added in the following steps.

12. The option to upload the questions from a CSV file is offered. If you accept this option, you will move to step 13. If you do not accept this option, you will move to step 20.
13. The option to download a base template to upload the questions in the CSV file is offered. If you accept this option, you move to step number 14; otherwise, you must go to step number 15.
14. A CSV template is downloaded to upload the questions, answers, and information for the game.
15. The CSV document must be uploaded to the platform.
16. The analysis and verification that the CSV file complies with the necessary characteristics, such as the extension, and the structure that is needed for each level are performed. As a result of this analysis, we obtain a true or false value.
17. The question is asked if the file is valid. If it is false, it moves to number 18. If it is true, it moves to number 27.
18. An error message is thrown, where it specifies what is failing in the system to upload the CSV file. When the message closes, we advance to number 19.
19. In the box presented in the graphical interface, correct the errors in the thematic content you want to upload to the platform. After this, it will advance to number 27.
20. Enter the question that will be added to the SG.
21. Enter the correct answer for the gamer for the question.
22. Enter the first incorrect answer for the question.

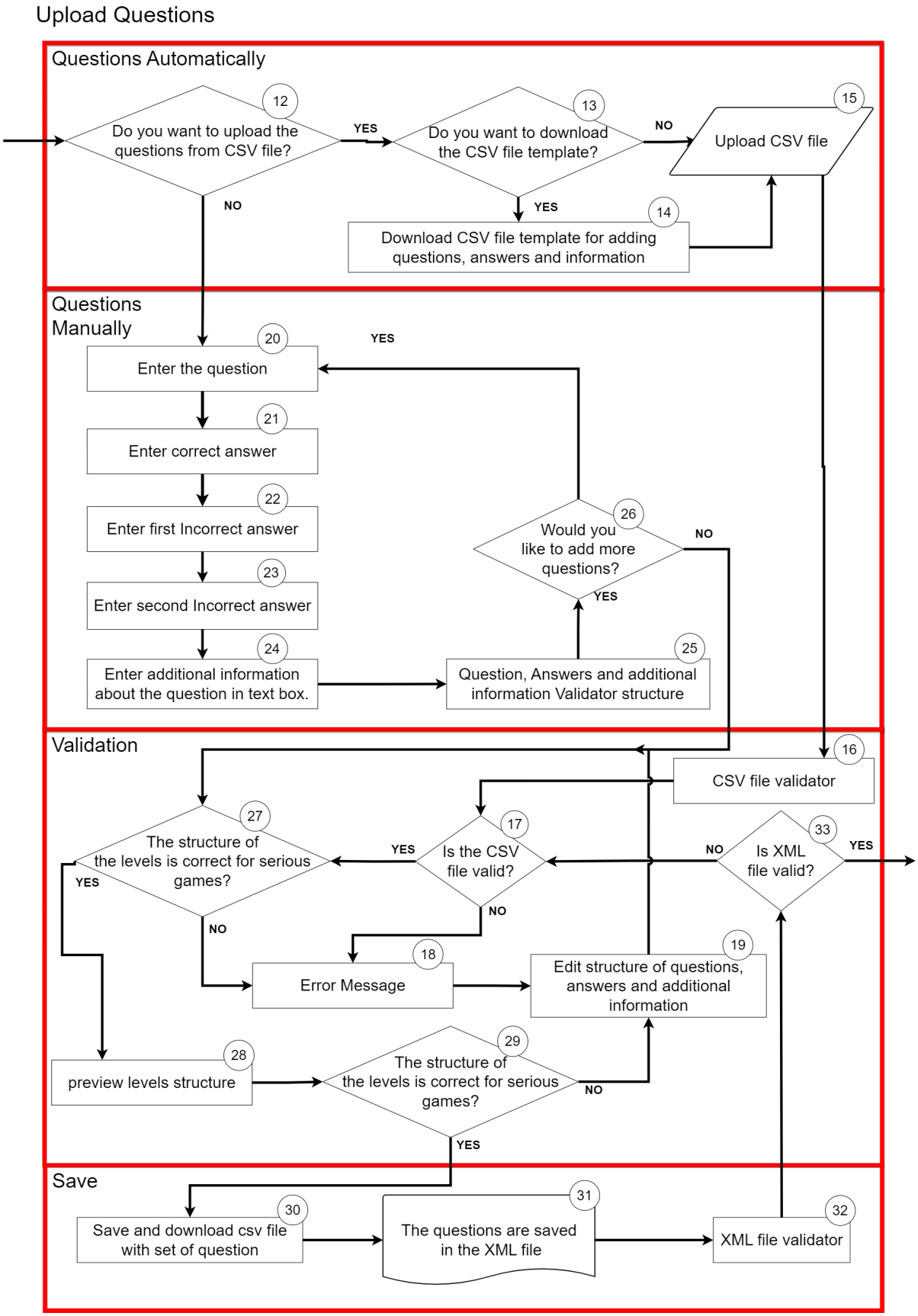

Figure 2. The second part of the flowchart is where you upload questions and answers.

23. Enter the second incorrect answer for the gamer.
24. Enter the information previous or post to the question asked.
25. A validation of the content is made, where the aspects for each question, answers, and didactic information are presented.
26. A confirmation is made whether you want to add more questions for the video game (as many as necessary). If your answer is true, go back to step number 20. If your answer is false, go to step number 27.
27. Validation is performed, where the question must have at least three answers and one correct answer. If each level has at least three questions and a maximum of five, continue to step number 28. If false, move to step number 18.
28. The levels are previewed with the questions and scenarios that are selected.
29. It must be confirmed if the user wants to add these questions to the SG. If the user confirms, it advances to number 30; if the answer is false, we go back to step number 19.
30. The CSV file is created, where the questions, answers, and information are saved. This file is available for the user to download.
31. The questions are added in the XML file.
32. It is validated that it was added correctly in the XML file. This validation gives us a true or false value.
33. A validation of the XML file is performed, which determines if there is any error in the XML file. If everything is saved, you will go to step number 34; otherwise, you will have to go to step number 18.

3.3. Phase C Source Code Generation

This is the last phase for the construction of the package that will be automatically built for display in the GDevelop video game engine. This package is built with the help of the XML file that was built during the selection and configuration process of the serious video game, as shown in the Figure 3.

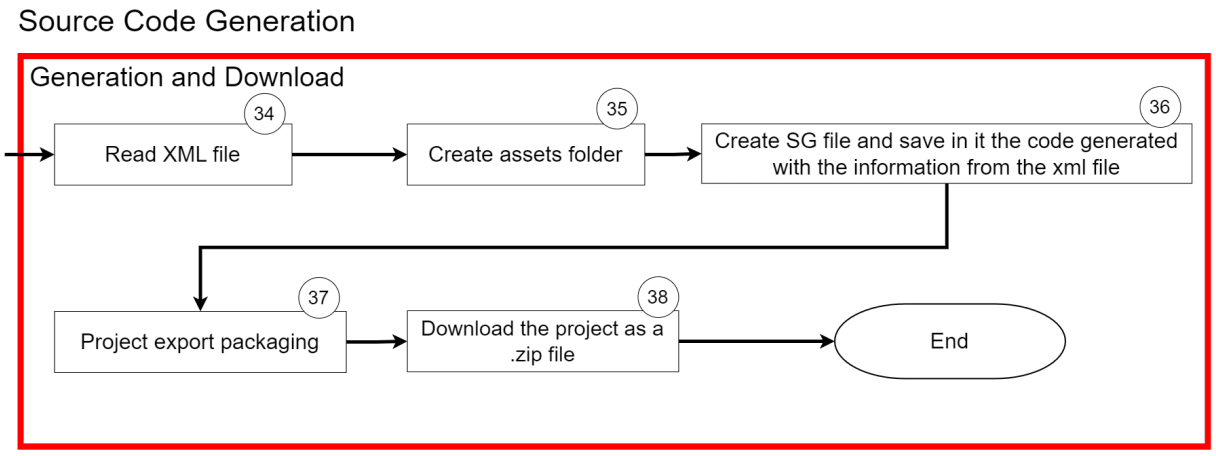

Figure 3. The flowchart's final section is where source code is generated.

In step number 33, it is validated that the XML file has the saved information to arrive at step number 34.

34. The XML file containing all the configurations of the SG and the educational content is read.
35. The creation of the folders containing the assets and the scenarios that will be used for the video game is done.
36. The XML file is sent to the SG document generator.
37. The files are packaged for download.
38. The assets package is downloaded, and the source code is ready to be uploaded to the Gdevelop game engine.

4. Use Case: Generate a Serious Game for Database Subject

The aim of the use case presented in this section is the development of a SG. Specifically, an inexperienced user needs to develop a SG to reinforce knowledge about a database course. Assuming that the teacher wants to generate a SG for the teaching of databases but the teacher does not have the knowledge for the development of a video game, nor does he have at his disposal an expert in video games who can support him in the process, he lacks the skills to synthesize the mechanics of a platform-type video game. The student desiring the SG to be developed should comply with the following characteristics:

1. Display the name of the SG before running it.
2. Make a platform game, such as Mario Bros.
3. Have each level have a different appearance theme.
4. Show the database course information.
5. Show questions that the learner must answer to continue in the video game.
6. Keep the student's score.
7. Tell a story to engage the learner.
8. Present a degree of difficulty for the learner.

For the development of this SG, the tool that implements the proposed model is used through a series of interfaces that facilitate the development of the video game. Next, we describe the set of actions necessary to develop a SG with the mentioned characteristics.

I. For the development of the SG using the tool proposed in this research, the first step is to name the video game. In this case, the SG was called "Adventure with relational algebra 2". Then, you can choose the main character for your SG. In this case, we chose the first option. Then, two selectors appear; in the first one from left to right, you can select the plot of the story in which you want to focus the SG. For this, the user chose the option "search plot". See Figure 4A.

II. In the SG generator tool, you can choose the number of levels that will integrate the video game. For this, the user chose that the video game will have three levels. The percentage bar will show the progress in the project configuration in total.

III. In the next screen, the user will be able to select one of the five scenarios available for the video game. As our user chose three levels, he will only be able to choose three of the five available scenarios, selecting scenario 1, scenario 3 and scenario 4. In Figure 4B, presents the interface.

IV. In the following screen, Figure 4C presents the interface. The user will have to upload the questions that he/she wants to appear throughout the video game; for this, he/she will have to have his/her question bank ready, and with this, he/she will be able to fill in box by box the corresponding question that he/she wants to be shown for the gamer user. The user will only be able to upload multiple choice questions and can choose the answer among three different options. In each level, you must upload at least three questions and a maximum of five. The user who uses our tool fills in the fields corresponding to the three levels, where they designate 5 questions for each one, giving a total of 15 questions. To these questions, the user will be able to attach information previous to the question to strengthen the knowledge.

V. In the last screen, you must confirm that everything is correct and is to the liking of the user, who is designing the SG to download the package that can be loaded into the game engine GDevelop. Once confirmed, our user downloads the package containing everything selected and the questions to be presented to the students. In Figure 4D, presents the interface.

VI. The software developer will unpack the content and will be able to view a JSON file and an assets folder containing all the elements of the game, which can be uploaded to GDevelop for a final review and then exported.

VII. Once downloaded, the developer must unpack the .zip file. In the folder, they will be able to visualize the JSON file and the folders with the assets that compose the video game in its totality.

VIII. When opening the Gdevelop platform, the file must be opened as a new project. You can test it in the Gdevelop platform. Once the developer evaluates it together with the teacher concerning the content and interaction, it can be exported to different platforms.

IX. Once exported, in Figure 5A, we can see the specific title and request by the user. In Figure 5B, we can find the summary of the story where we put into context the objective to be achieved and the motivation of the player to enter the video game. In Figure 5C, we can see how educational reinforcement information is presented to the user and the questions to reinforce that knowledge.

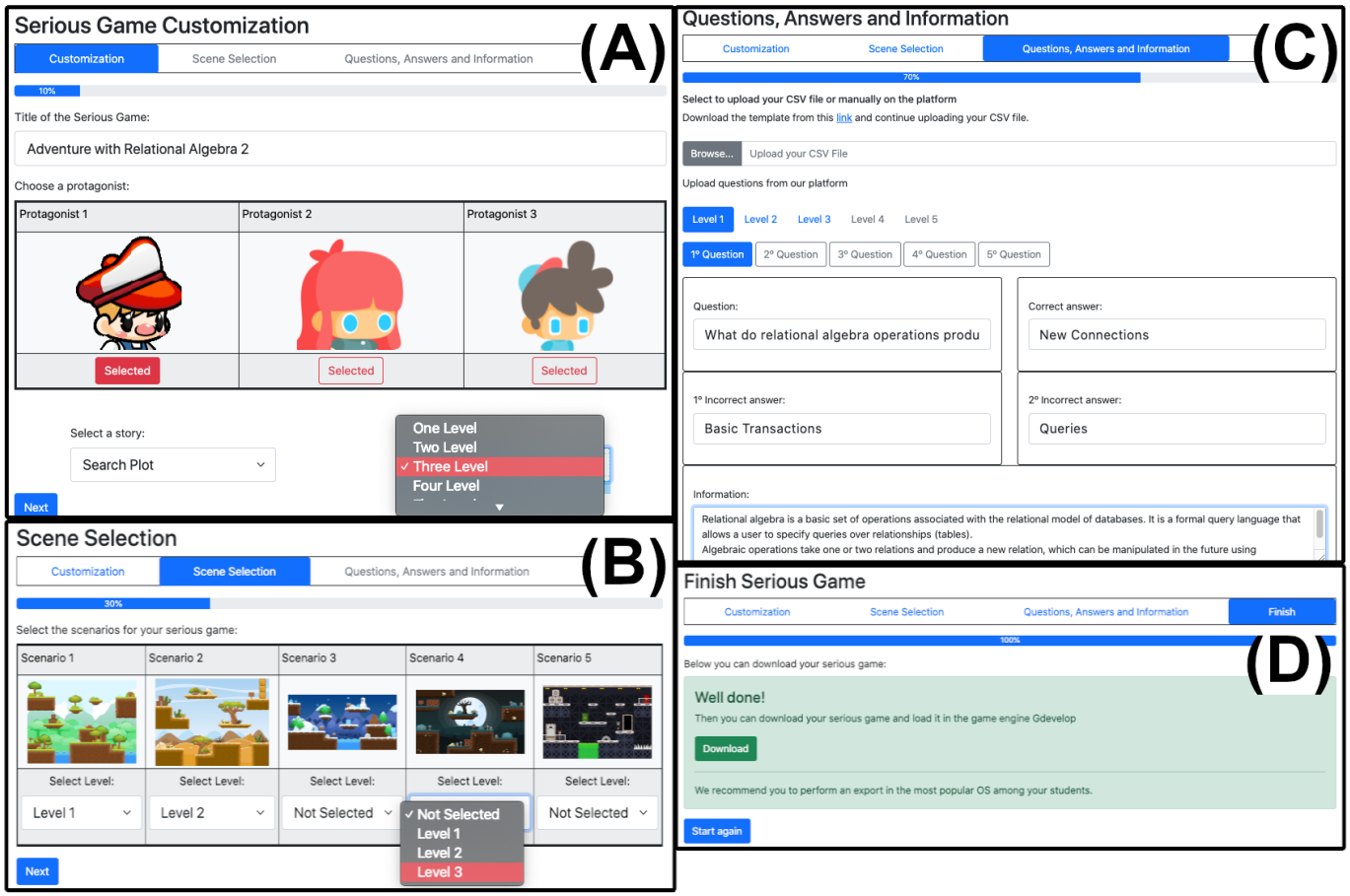

Figure 4. In (**A**), SG customization takes place, in (**B**) levels are selected, in (**C**) questions and answers are loaded, and in (**D**) there is a graphical interface for downloading the file.

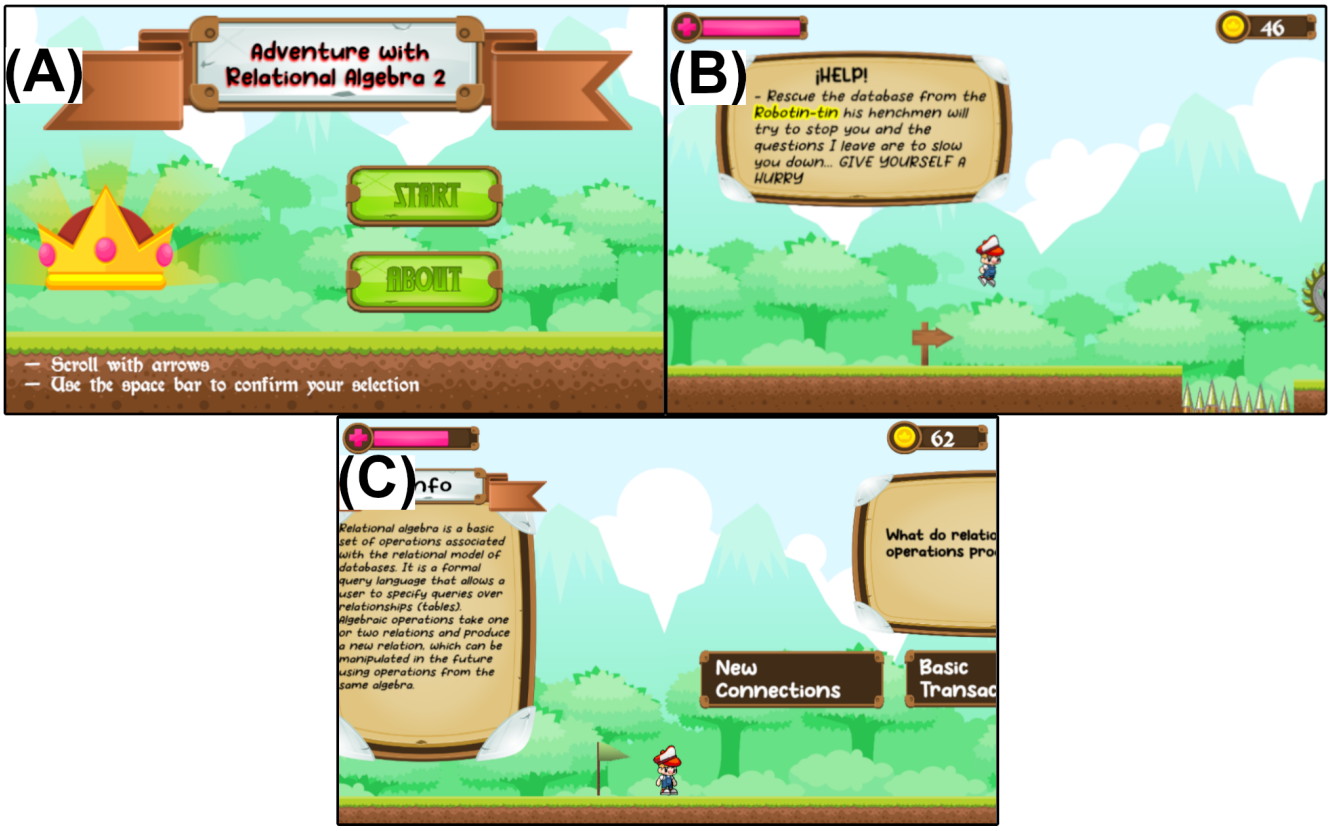

Figure 5. Some of the graphical interfaces of the SG are (**A**) the main menu; (**B**) the objective of the video game is presented; and (**C**) educational information and a question for the end user are presented.

5. Evaluation and Discussion

The literature on software engineering reports several methods for software tool assessment. According to Kitchenham et al. [36], these methods can be classified into quantitative, qualitative, and hybrid. This investigation proposes quantitative and qualitative types of evaluation. For our research and evaluation, a mixed-type evaluation is selected. The evaluation design is depicted in Figure 6. We selected two different types of evaluations for the SG that is generated by the tool.

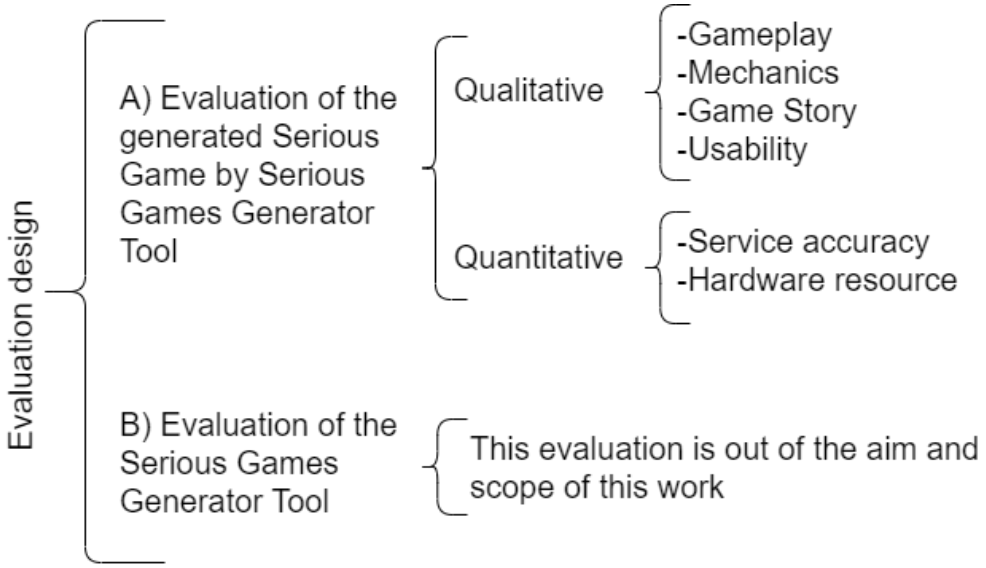

Figure 6. Evaluation design.

5.1. Video Games Evaluations

There are many heuristics in the field of video games, some of which are repetitive and others so isolated that they produce contradictory results. The main purpose of this evaluation is to define usability for the SG generated by the tool. Existing heuristics serve to understand weaknesses and strengths. We decided to evaluate two aspects of the project: the SG developed within the application and its evaluation using a qualitative approach focused on usability. This evaluation will be possible with the help of a group of students who will answer the evaluation presented in the article by Quinn et al. [37] to perform a statistical analysis of the final results.

Some of the types of assessments that have been conducted to evaluate the qualitative part of SGs will be mentioned below. To evaluate a SG, the evaluation proposals available in various research were analyzed in the authors' research evaluation proposal. Quinn et al. [37] presented a methodology of data collection after the evaluation to provide detailed information about the players, including such details as gameplay, mechanics, game history, and usability.

Another proposal is from the authors Fu et al. [38]. They proposed a scale that effectively measures the satisfaction offered by e-learning games to help the game designer understand the strengths and weaknesses of the game effectively from the learner's point of view by evaluating, in particular, the domains of concentration, goal clarity, feedback, challenge, autonomics, immersion, social interaction, and knowledge enhancement.

Meanwhile, the authors of [39] presented a new instrument that works to measure video game satisfaction, called the game user experience satisfaction scale (GUESS), with nine subscales. The GUESS demonstrated content validity, internal consistency, and convergent and discriminant validity. GUESS focuses on items such as usability/playability, narrative, absorption, fun, creative freedom, audio, gratifications, connectivity, and visual aesthetics. Another proposal found was that of the authors of [40], who proposed the evaluation of playability through different methods, which is one of the main themes of player experience in video games. They presented a new approach to evaluating PX using educational playability. They considered important the points of intrinsic gameplay, mechanical gameplay, interactive gameplay, artistic gameplay, and interpersonal education.

A form of evaluation that was investigated is the one proposed by the authors of [41]. In the proposal, they present a game experience questionnaire. It has questionnaires with questions and ranges of feelings concerning the video game, concentrating particularly on absorption, flow, presence, and immersion.

Justifying the choice within the characteristics of the selection of the type of evaluation, we decided to select the SG evaluation methodology presented in the research of Quinn et al. [37]. We consider that it presents a questionnaire that meets the main aspects that a SG should have. Furthermore, it is very light and concrete for users to know their perspectives on the SG presented to them. This methodology uses focus groups. Data-collection methodologies were used to obtain detailed information from the players, which was then analyzed and used to introduce continuous and timely changes to the game.

For a brief description of the first evaluation of the SG generated, it consisted of a survey that has 22 questions presented on a Likert scale. The survey is presented in a form created in Google forms. The SG was on the website https://liluo.io/pedrosilva/version-base-de-datos (accessed on 10 November 2022) for users to perform the evaluation remotely.

5.2. ISO Evaluations

The following are some of the types of evaluations that have been performed to assess the quantitative part of SGs (ISO). Software quality management can be classified into two types. The first type assumes that software organizations can produce quality products if their development methods and procedures are applied. Therefore, although it is a viewpoint that emphasizes the evaluation of the quality of the development process itself, proper development methods and procedures do not guarantee product quality. Therefore, the second perspective emphasizes the evaluation of product quality. Based on these

observations, two software quality assessment models have been developed and applied in the IEEE [42].

ISO/IEC 25000 covers procedures for establishing evaluation requirements, specifications, design, and evaluation practices. The use of this standard model is suitable for developers, purchasers, and assessors to evaluate the quality of software products. ISO/IEC 25000 is a software evaluation model to integrate and enhance existing international standards developed to replace ISO/IEC 9126 [43].

The following are some of the types of assessments that have been performed to evaluate the quantitative part of SGs. ISO/IEC 25000 provides guidance for the use of the new series and international standards, called software product quality requirements and assessment. The standards that make up this division define common models, terms, and definitions, while ISO/IEC 2501n corresponds to the quality model, where the characteristics for internal, external, and usage quality are detailed. ISO/IEC 2502n corresponds to quality measurement and includes a software product quality reference model, mathematical definitions of quality metrics, and a practical guide for their application. ISO/IEC 2503n is oriented to help in the specification of requirements for a software product to be developed or as input for an evaluation process. Finally, ISO/IEC 2504n provides requirements, recommendations, and guidelines for the evaluation of a software product.

The ISO/IEC 25000 model defines eight characteristics for the internal and external quality of a software product: suitability, functionality, reliability, performance efficiency, usability, security, compatibility, maintainability, and portability. For our research, we selected performance efficiency, which evaluates the ability of a software product or system to provide adequate performance for the number of resources used under given conditions. This characteristic is divided into the following sub-characteristics: Temporal behavior: the ability of a software system to provide adequate response and processing times. Resource utilization: the ability of a software system to use appropriate amounts and types of resources. Capacity: the ability of a software system to satisfy given requirements. Portability: the ability of a software system or component to be moved from one environment to another without affecting the functionality of each system. This characteristic is subdivided into the following sub characteristics: Adaptability: the ability of a software system to adapt to different environments. Installability: the ability of a system to be easily installed and/or uninstalled. Substitutability: the ability of a software system to be used in place of another system in the same environment and for the same purpose. The quality of use model defines five characteristics: effectiveness, efficiency, satisfaction, absence of risk, and context coverage. These, in turn, are subdivided into sub-characteristics that can be measured with the quality of useful metrics. The result of the quality of use necessarily depends on the achievement of external quality, which in turn necessarily depends on the achievement of internal quality. Effectiveness: the ability of the software system to achieve the user's objectives or needs when using the system. Efficiency: the ability of the software system to achieve the user's objectives, using minimal resources [43].

5.3. Qualitative Evaluation

The design of the qualitative evaluation and the results obtained are presented below.

5.3.1. Evaluation Design

For this research, a survey structured in two parts is presented. The first part describes the users of video games and knows their particularities, as well as their preferences in video games. For the second part of the survey, we considered carrying out the guide presented by Quinn et al. [37] since it seems to us the most accurate regarding the objective of this evaluation, which is to know the strengths and weaknesses of the SG generated by the tool, focusing on the four main areas of this evaluation: gameplay, mechanics, a game story, and usability. As shown in Figure 6, there were two options for evaluating the SG development tool. The first option was to evaluate the tool with a survey based on heuristics for SGs, and the second alternative was to evaluate the generated applications.

In the end, we decided to evaluate the SG generated by the tool. This evaluation was divided in two. The second evaluation will touch on the performance of the SG when installed on a computer and if it consumes many services of the computer where it is used.

A group of students were asked to respond to the qualitative evaluation of our SG. The group consisted of forty university students aged 19 to 32. All of them are students in the formation of careers of the faculty of statistics and computer science, as follows: a Bachelor's degree in computer technologies, a Bachelor's degree in networks and computer services, and a Bachelor's degree in software engineering in Universidad Veracruzana. In the same way, students of the engineering career in computer systems of the weekly and Saturday shifts that belong to the Instituto Tecnologico Superior de Teziutlan were also included since in all of them, the subject of databases is taught. The main topic of the SG is the theory of algebraic relations that is presented for database development. In particular, they had to have already passed the subject so that it would not be complicated to perform the course in the SG presented.

To evaluate the SG, the students first interacted with the graphical interface of the game. After that, they had to perform the game tutorial, where all possible interactions they could encounter during the game were explained to them. After this interaction, they could find the level where the knowledge questions would be asked. The development of the SG was carried out with the help of the teacher who specialized in the subject of databases and gave us a battery of questions that we could consider for the SG. The development of the personnel and scenario designs was downloaded from https://www.gameart2d.com/freebies.html (accessed on 10 November 2022), which has a Creative Common Zero (CC0), i.e., public domain license, and the development of the SG was carried out by the authors of this research work.

Once the experience with the SG was over, the users had to answer the questionnaire as honestly as possible. Once the evaluation was finished, they could leave the evaluation room.

5.3.2. Results

Once the survey was conducted with 75 students, who interacted with the SG and answered the survey, a detailed analysis of the users was carried out.

Figure 7 depicts that more than half of the users are video game players, and 27 more consider themselves to be part-time gamers. Regarding the ages that we were able to capture for the survey, students were between 21 and 22 years old. This was because the students had already taken a database course and were in higher semesters.

In the gender graph, we can see that at least 76% of the respondents are men, after subtracting the 24% of women. The PC is the most popular device, with 41% acceptance and first place in the survey, followed by smartphones with 36% and consoles with 19%.

Another important question in the survey was whether users had interacted with any SG, to which 35 respondents confirmed that they had, 26 students answered no, and 14 people were not sure if their interaction with an app could be considered a SG. This is because the objective of the app was not clear or was never presented to the end user.

They were also asked if they play video games alone or with company, and 56% responded that they play alone, while 38.7% said they prefer to play with their friends. They were asked about their time spent playing video games, and 31 students answered that they spend less than 1 hour on this activity, while 27 students spend 1 to 2 hours a day, and 16 students spend at least 3 to 4 hours a day.

As we have managed to describe the users who took the survey, we will move on to analyze our main objective, which is to evaluate the aspects of our video game, such as gameplay, mechanics, a game story, and usability.

For the item of gameplay, in the first question, if we made clear the objectives of the game, we obtained a good response, with 31 students being in total agreement and 40 being in agreement with the objective of the game. For the second question, if we present the progress of the video game in the opinions, we need to work on this point. Another

important question, number 4, is whether the challenge, strategy, and rhythm are balanced. We had a high acceptance rate considering that 38 students answered that they agreed and 23 agreed.

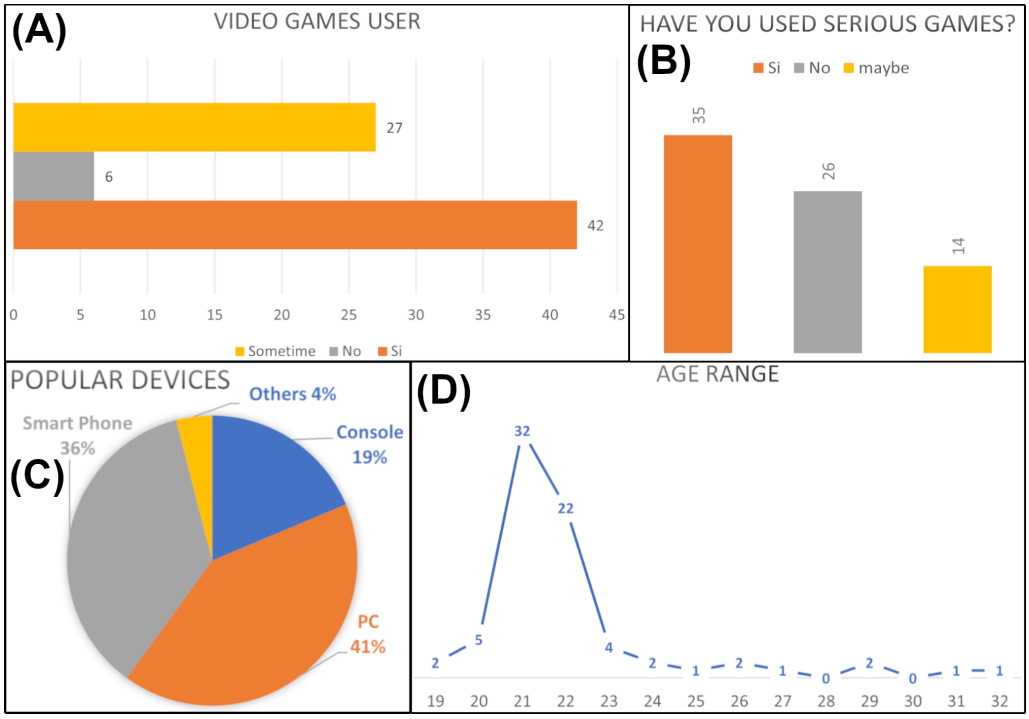

Figure 7. The data from the survey is presented in the graphs below for: (**A**) video game users; (**B**) the number of users who have used a SG; (**C**) the most popular video game devices among our respondents; and (**D**) the age range of the interviewed users.

In the question of whether their first experience was encouraging, 17 answered that they agreed that it was encouraging, and 36 agreed that it was encouraging.

In the area of game mechanics, 92% said that the game mechanics are consistent, 94% agreed that the controls are easy to learn and that the navigation of the game is very easy, and 89% agreed that the exploration of video games is very easy.

We should work on and improve the game story because, while we received 66% acceptance in the question "Does the game story make sense?" the rest did not understand its meaning of it, so we will have to change it and present it differently. When asked if the tasks were repetitive or boring, 42% of the respondents said they neither agreed nor disagreed. This is a very high score, so it is another point that needs to be improved. On the question "Can he express himself?" we have a high percentage of neither agree nor disagree, with 45%. The points where we scored well are in the relationship between the character and the player, with 39 students agreeing that it is a good relationship. In the last question of the game story area, we obtained 64% acceptance of the playability of the story.

We had strong opinions in the area of usability, as 66 students rated it as coherent in the first question. In terms of whether it is similar to other games, we obtained the acceptance of 66 students. On the question of whether the information that the user receives from the game is adequate, we obtained a response from 71 students who accepted it. On the question of whether the user receives all the necessary information, we received an acceptance response from 65 students. When asked about the efficiency of the screen, and whether it is visually appealing, 59 students said that they agree. In the questions that talk about the visual and auditory, we had a response of acceptance, with 67 and 62, respectively, confirming it. This is illustrated in Table 1 below.

Table 1. Table of questions for the SG.

Heuristic	Description	Totally Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree
Gameplay	Does the game provide clear goals for the user?	41%	53%	5%	0%	0%
	Does the player see the progress in the game?	20%	52%	16%	12%	0%
	Does the player feel in control of the game?	33%	49%	12%	4%	1%
	Are the challenge, strategy, and pace balanced?	31%	51%	9%	9%	0%
	Was the first-time experience encouraging?	23%	48%	20%	7%	3%
Mechanics	Are the game mechanics consistent throughout the game?	32%	60%	7%	1%	0%
	Are the controls easy to learn?	60%	35%	4%	1%	0%
	Does the navigation system support the ease of gameplay?	31%	55%	8%	7%	0%
	Is it easy to explore the playfield?	33%	56%	9%	1%	0%
Game Story	Is the game story meaningful?	15%	52%	25%	7%	1%
	Are there repetitive or boring tasks?	8%	32%	43%	16%	1%
	Does the player have the opportunity to express him/herself?	7%	29%	45%	13%	5%
	Does the player relate to the characters?	4%	48%	25%	17%	5%
	Did the gameplay make sense with the story?	12%	52%	27%	7%	3%
Usability	Is the user interface consistent throughout the game?	27%	61%	11%	1%	0%
	Is the user interface similar to other games the user has experienced?	39%	49%	9%	3%	0%
	Is the feedback to the user from the game adequate?	41%	55%	4%	0%	0%
	Is all information that the user needs displayed clearly when the user needs it?	37%	49%	8%	5%	0%
	Is the screen layout efficient?	29%	52%	13%	4%	1%
	Is the screen layout visually appealing?	25%	53%	16%	5%	0%
	Does the visual appearance support the playing of the game?	28%	61%	7%	3%	1%
	Do the audio effects support the playing of the game?	28%	55%	16%	1%	0%

In their investigation, Symborski et al. [44], they discuss the use of iterative evaluation in the design and development of SGs. The authors argue that traditional methods of evaluating SGs, such as randomized controlled trials, are often not sensitive to the unique characteristics of SGs. Instead, they propose using an iterative evaluation approach, in which the game is evaluated and revised multiple times throughout the development process. The investigation by [44] describes a case study in which this approach was used to design a SG for training healthcare professionals. The results of the study indicate that this approach can lead to more effective and engaging SGs.

Evaluating SGs is crucial to ensuring that they are effective in achieving their intended learning objectives and delivering their intended message. Additionally, evaluating a SG allows for the identification of areas for improvement and the making of necessary adjustments to enhance the game's effectiveness. Furthermore, it is important to compare the results of the evaluations with other related studies to show the advantages of the game and to demonstrate the effectiveness of the game in achieving the objectives. Overall, evaluating SGs is an essential step in the design and development process to ensure that they are effective in achieving their intended goals.

Evaluating the usability, mechanics, gameplay, and story of a SG is essential to ensuring that the game is effective in delivering its intended message and achieving its intended learning objectives. Usability is critical, as it ensures that the game is easy for players to interact with and perform the intended tasks. A game that is not user-friendly can be frustrating for players and negatively impact their motivation and engagement in the

learning process. Mechanics are the underlying rules and systems that govern the game's behavior. Well-designed mechanics can help to create a sense of immersion and engagement for the players, which is crucial for a SG.

Gameplay refers to the overall experience of playing the game and includes elements such as pacing, challenge, and progression. A well-designed game can provide a balance of challenge and enjoyment for the players, which will keep them engaged and motivated. The story of the game is also important, as it can help to create a sense of immersion and engagement for the players and also help to convey the message or learning objectives of the game. Overall, evaluating the usability, mechanics, gameplay, and story of a SG is crucial to ensuring that the game is effective in delivering its intended message and achieving its intended learning objectives. It also helps to improve the player's overall engagement and motivation to play and learn.

5.4. Quantitative Evaluation

The quantitative evaluation design, implementation, and outcome of the evaluations are presented below.

5.4.1. Evaluation Design

To carry out the evaluations in this area, we considered the process descriptions presented in the ISO/IEC 25023 standard, which considers internal and external quality metrics for performance efficiency, focusing particularly on the aspect of time behavior for the development of the evaluation, which is explained in the following table, as well as the procedures we followed to carry it out.

To design the evaluation of the SG, the game itself was exported to a specialized web page for video games designed in Gdevelop. The following link shows the SG https://liluo.io/pedrosilva/version-base-de-datos (accessed on 10 November 2022) that was evaluated. Once the game was on the corresponding web page, the tests were carried out on computer equipment with a processor model AMD Ryzen 5 3400 G, a memory ram of 8 GB, and a hard disk of solid state of 500 GB, a connection to the Internet of 40 MB and a Windows operating system. All applications running in the foreground on the PC were closed to lower the CPU performance to the minimum to proceed to run the evaluation, with the help of the Windows task manager. The software of navigation called Google Chrome was executed; after this, we opened a tab of the navigator the link to the SG. To carry out the evaluation, a user made a tour of the video game, while the movement of the task manager was observed.

From the task manager that Windows has, we were able to obtain the results to carry out the operations presented in the following Table 2.

Table 2. Evaluation table for ISO/IEC 25030 resources utilization.

Sub Characteristics	Metrics	Purpose	Method of Application	Formula	Desired Value
Resource utilization	CPU utilization	How much CPU time is used to perform a given task?	Take the operation time and the amount of CPU time used to perform a task	$x = B - A$ A = The amount of CPU time that is actually used to perform a task B = Operation time Where: $B > 0$	$0 \leqq x \leqq 1$ Closest to 0 is the best where the worst case is $\geqq$15t
	Memory Utilization	How much memory space is used to perform a given task?	Measure the total amount of memory space and the amount of memory space that is actually used to perform a task	$x = B - A$ A = Number of memory spaces that is actually used to perform a task B = Total amount of memory spaces Where: $B > 0$	$0 \leqq x \leqq 15$ The closest to 0 is the best

The corresponding information from the evaluation of the SG, which was carried out with ISO/IEC 25030, was captured, and we will present the results in the following section.

5.4.2. Results

The performance of the CPU is observed to be at its lowest at the start of the evaluation, as seen in Figure 8. The browser software Google Chrome was run and a browser tab was opened with the link to the SG. The game was initiated with the assistance of a user utilizing the computer keyboard.

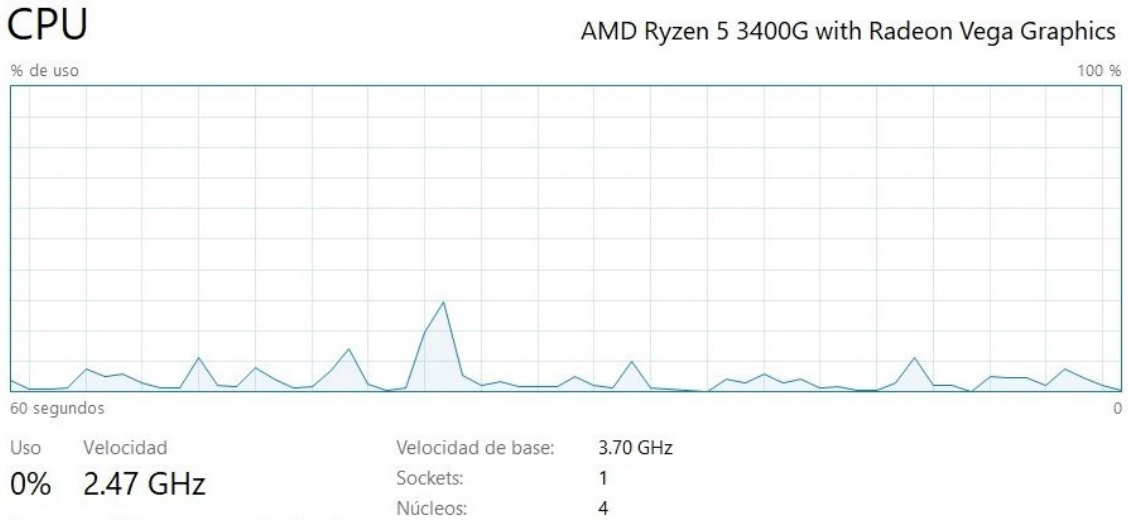

Figure 8. The CPU performance is lowest.

The use of the game increased the CPU work at a minimum, but as the game was fully loaded in the browser, the work time decreased until it stabilized at a certain percentage and did not move, as shown in Figure 9.

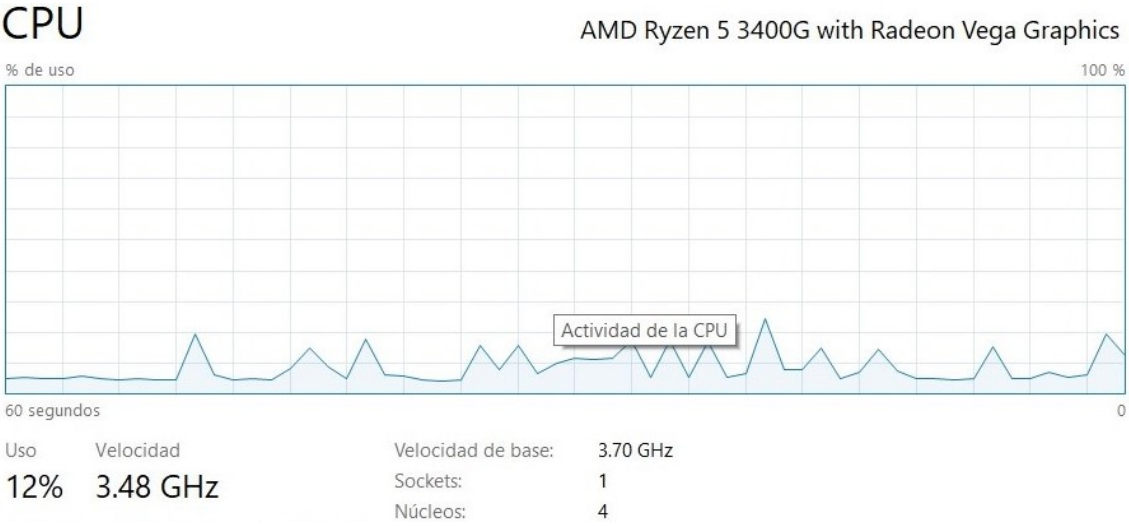

Figure 9. The CPU performance increased

With these data, we went on to perform the corresponding formulas for the calculation of the evaluation methods that we took from ISO/IEC 25023.

To gather information about the CPU utilization, data was collected and used to complement arithmetic operations. The utilization of the data was calculated by determining the amount of CPU time used to perform a task (A) and the operation time (B).

$$A = 154, B = 164, \quad (1)$$

$$x = 154 - 164, \tag{2}$$

$$x = 10, \tag{3}$$

The conclusion of the evaluation is clear in saying that "the closest to 0 is the best, while the worst case is ≧15t". In this case, being at 10, we are within an acceptable range for the use of the application.

Given that the second point we wanted to know was the memory using the data collected, we used the following formula to supplement our arithmetic operation, where A = the number of memory spaces that are actually used to perform a task and B = the total number of memory spaces:

$$A = 3.3 \text{ GB}, B = 3.8 \text{ GB}, \tag{4}$$

$$x = 3.8 - 3.3, \tag{5}$$

$$x = 0.5 \text{ GB}, \tag{6}$$

The conclusion of the evaluation is clear in saying that "The closest to 0 is the best, while the worst case is ≧15t". In this case, being at 0.5, we are within a very narrow range, so it is impeccable memory management, an advantage of the use of the application on a web site since everything is on the server side.

5.5. Discussion

The two evaluations provide valuable insights into the performance of the SG. Based on the qualitative feedback from users, the development team is able to identify any shortcomings and make necessary improvements to enhance the user experience. On the other hand, the quantitative evaluation results demonstrate that the SG meets the stringent standards set by ISO/IEC 25023 for entertainment software.

However, there is room for improvement in the area of game storytelling. The results show that the story content and presentation fell below expectations and need to be reworked in order to engage and motivate players. The feedback collected from the suggestion box will also be taken into consideration.

Overall, the evaluations indicate that the SG is well-received by students and has promising characteristics. Further efforts will be made to refine and improve the design and user experience.

The findings from the two evaluations showed both positive and negative aspects of the SG. On one hand, the evaluation process involved the participation of users, who provided their qualitative perspectives on the game. This helped to identify any shortcomings and deficiencies in the game's interaction, which can then be improved to meet the expectations of the final users, in this case, students.

In the quantitative evaluation, the results were satisfactory as the development of the SG met the strict standards set by ISO/IEC 25023 for evaluating entertainment software. However, there was room for improvement in the area of game storytelling, as the results were below expectations. To address this, adjustments need to be made to the content and presentation of the story in order to better engage the player and make the story more interesting.

Finally, feedback from users was collected through a suggestion box and will be taken into consideration for future improvements to the design and functionality of the SG. Overall, the SG has encouraging and acceptable characteristics for the surveyed students, and there is potential for further improvement.

6. Conclusions

A model was developed to create SGs that consider the characteristics of commercial video games In general, video games are known for their art, mechanics, story, and development technology. To incorporate these elements in the creation of SGs, a popular and easy-to-understand game mechanic was selected: platforming. The result was a

combination of platform game mechanics and trivia elements, with a focus on story. The utilization of SGs in the classroom should serve to reinforce student knowledge as intended by the teacher. This model aims to assist software developers in creating SGs by reducing development time and increasing familiarity with their components. The developed tool supports the creation of SGs and expedites their development process. The evaluations of the resulting SG were satisfactory, with room for improvement based on our observations. In the future, the proposed tool should be evaluated by both usability experts and teachers to confirm its benefit to the end users.

In addition, it is also important to consider the potential impact that this tool may have on the learning process itself. The use of SGs in the classroom can provide an engaging and interactive learning experience, which has been shown to be effective in reinforcing knowledge and helping students to retain information. By streamlining the creation process of SGs, it is possible to encourage their wider adoption in the classroom. Also, it is essential to consider the potential for this tool to be used beyond the classroom, in alternative educational settings. SGs can be used to reinforce learning for a wide range of subjects, and can be tailored to suit specific training requirements.

Finally, even though our tool is focused on streamlining the process of creating SGs, it still strives to maintain high standards of design, development, and evaluation. The creation of SGs must always take into consideration the educational objectives, the target audience, and the goals of the project. The tool should be viewed as a means to support the creation process, not as a replacement for careful planning, design, and evaluation. In conclusion, the development of this tool for the creation of SGs holds significant potential for improving the learning experience for students by reinforcing their knowledge in a fun and engaging way. However, it is important to both consider the potential impact of this tool and to continue maintaining high standards of design and evaluation. The latter to ensure that SGs are effective and meet the needs of their intended audience.

Author Contributions: Conceptualization, P.O.S.-V., V.Y.R.-M. and E.B.-G.; Data curation, E.B.-G. and C.M.-G.; Formal analysis, P.O.S.-V. and G.A.-H.; Funding acquisition, E.B.-G., C.M.-G. and L.G.M.-J.; Investigation, P.O.S.-V. and V.Y.R.-M.; Methodology, V.Y.R.-M. and G.A.-H.; Project administration, G.A.-H.; Resources, G.A.-H.; Software, P.O.S.-V. and V.Y.R-M.; Supervision, V.Y.R.-M. and G.A.-H.; Validation, C.M.-G. and E.B.-G.; Visualization, L.G.M.-J.; Writing—original draft, P.O.S.-V.; and Writing—review and editing, V.Y.R.-M. and E.B.-G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially developed under the support of the National Council of Science and Technology (CONACYT-Mexico) in the scope of the project "Infraestructura para Agilizar el Desarrollo de Sistemas Centrados en el Usuario" (Cátedras, Ref. 3053). In addition, the authors thank CONACYT for the doctoral scholarship (number 395377) granted to the first author. We also the Universidad Veracruzana for the support in the development of this research.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy concerns of study.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Hanes, L.; Stone, R. A model of heritage content to support the design and analysis of video games for history education. *J. Comput. Educ.* **2019**, *6*, 587–612. [CrossRef]
2. Catalano, C.E.; Luccini, A.M.; Mortara, M. Guidelines for an effective design of serious games. *Int. J. Serious Games* **2014**, *1*. [CrossRef]
3. de Lope, R.P.; López Arcos, J.R.; Medina-Medina, N.; Paderewski, P.; Gutiérrez-Vela, F.L. Design methodology for educational games based on graphical notations: Designing Urano. *Entertain. Comput.* **2017**, *18*, 1–14. [CrossRef]

4. González García, C.; Núñez-Valdez, E.R.; Moreno-Ger, P.; González Crespo, R.; Pelayo G-Bustelo, B.C.; Cueva Lovelle, J.M. Agile development of multiplatform educational video games using a Domain-Specific Language. *Univers. Access Inf. Soc.* **2019**, *18*, 599–614. [CrossRef]
5. Chamberlin, B.; Trespalacios, J.; Gallagher, R. The learning games design model: Immersion, collaboration, and outcomes-driven development. *Int. J. Game-Based Learn.* **2012**, *2*, 87–110. [CrossRef]
6. Mostafa, M.; Faragallah, O.S. Development of Serious Games for Teaching Information Security Courses. *IEEE Access* **2019**, *7*, 169293–169305. [CrossRef]
7. Zarraonandia, T.; Diaz, P.; Aedo, I. Using combinatorial creativity to support end-user design of digital games. *Multimed. Tools Appl.* **2017**, *76*, 9073–9098. [CrossRef]
8. Aleem, S.; Capretz, L.F.; Ahmed, F. Critical Success Factors to Improve the Game Development Process from a Developer's Perspective. *J. Comput. Sci. Technol.* **2016**, *31*, 925–950. [CrossRef]
9. Ahmad, A.; Law, E.L.C. Educators as Gamemasters: Creating Serious Role Playing Game with "aRQS". *Proc. ACM Hum. Comput. Interact.* **2021**, *5*, 230–259. [CrossRef]
10. Marchiori, E.J.; del Blanco, Á.; Torrente, J.; Martinez-Ortiz, I.; Fernández-Manjón, B. A visual language for the creation of narrative educational games. *J. Vis. Lang. Comput.* **2011**, *22*, 443–452. [CrossRef]
11. Marne, B.; Wisdom, J.; Huynh-Kim-Bang, B.; Labat, J.M. The Six Facets of Serious Game Design: A Methodology Enhanced by Our Design Pattern Library. In Proceedings of the European Conference on Technology Enhanced Learning, Saarbrücken, Germany, 18–21 September 2012; pp. 208–221. [CrossRef]
12. Carvalho, M.B.; Bellotti, F.; Berta, R.; De Gloria, A.; Sedano, C.I.; Hauge, J.B.; Hu, J.; Rauterberg, M. An activity theory-based model for serious games analysis and conceptual design. *Comput. Educ.* **2015**, *87*, 166–181. . [CrossRef]
13. Zarraonandia, T.; Diaz, P.; Aedo, I.; Ruiz, M.R. Designing educational games through a conceptual model based on rules and scenarios. *Multimed. Tools Appl.* **2015**, *74*, 4535–4559. [CrossRef]
14. Arnab, S.; Lim, T.; Carvalho, M.B.; Bellotti, F.; De Freitas, S.; Louchart, S.; Suttie, N.; Berta, R.; De Gloria, A. Murdoch Research Repository Mapping Learning and Game Mechanics for Serious Games Analysis. *Br. J. Educ. Technol.* **2015**, *46*, 391–411. [CrossRef]
15. Cano, S.; Munoz Arteaga, J.; Collazos, C.A.; Gonzalez, C.S.; Zapata, S. Toward a methodology for serious games design for children with auditory impairments. *IEEE Lat. Am. Trans.* **2016**, *14*, 2511–2521. [CrossRef]
16. Orji, R.; Mandryk, R.L.; Vassileva, J. Improving the efficacy of games for change using personalization models. *ACM Trans. Comput. Hum. Interact.* **2017**, *24*, 32–54. [CrossRef]
17. Andreoli, R.; Corolla, A.; Faggiano, A.; Malandrino, D.; Pirozzi, D.; Ranaldi, M.; Santangelo, G.; Scarano, V. A framework to design, develop, and evaluate immersive and collaborative serious games in cultural heritage. *J. Comput. Cult. Herit.* **2017**, *11*, 4–26. [CrossRef]
18. Heintz, S.; Law, E.L. Digital educational games: Methodologies for evaluating the impact of game type. *ACM Trans. Comput. Hum. Interact.* **2018**, *25* 8–33. [CrossRef]
19. Carrión-Toro, M.; Santorum, M.; Acosta-Vargas, P.; Aguilar, J.; Pérez, M. iPlus a user-centered methodology for serious games design. *Appl. Sci.* **2020**, *10*, 9007. [CrossRef]
20. Pérez, F.; Lapeña, R.; Marcén, A.C.; Cetina, C. Topic modeling for feature location in software models: Studying both code generation and interpreted models. *Inf. Softw. Technol.* **2021**, *140*, 106676. [CrossRef]
21. Blasco, D.; Font, J.; Zamorano, M.; Cetina, C. An evolutionary approach for generating software models: The case of Kromaia in Game Software Engineering. *J. Syst. Softw.* **2021**, *171*, 110804. [CrossRef]
22. Oren, M.; Pedersen, S.; Butler-Purry, K.L. Teaching Digital Circuit Design with a 3-D Video Game: The Impact of Using In-Game Tools on Students' Performance. *IEEE Trans. Educ.* **2021**, *64*, 24–31. [CrossRef]
23. Göbel, S.; Salvatore, L.; Konrad, R. StoryTec: A digital storytelling platform for the authoring and experiencing of interactive and non-linear stories. In Proceedings of the 4th International Conference on Automated Solutions for Cross Media Content and Multi-Channel Distribution, Florence, Italy, 17–19 November 2008; pp. 103–110. [CrossRef]
24. Mourato, F.; Dos Santos, M.P.; Birra, F. Integrated system for automatic platform game level creation with difficulty and content adaptation. *Lect. Notes Comput. Sci.* **2012**, *7522*, 409–412. [CrossRef]
25. Mourato, F.J.D.S.V. Enhancing Automatic Level Generation for Platform Videogames. Ph.D. Thesis, Universidade NOVA de Lisboa, Lisboa, Portugal, 2015.
26. McArthur, V.; Teather, R.J. Serious mods: A case for modding in serious games pedagogy. In Proceedings of the 2015 IEEE Games Entertainment Media Conference, Toronto, ON, Canada, 14–16 October 2015. [CrossRef]
27. Perez-Colado, I.J.; Perez-Colado, V.M.; Martinez-Ortiz, I.; Freire-Moran, M.; Fernandez-Manjon, B. UAdventure: The eAdventure reboot: Combining the experience of commercial gaming tools and tailored educational tools. *IEEE Glob. Eng. Educ. Conf.* **2017**, *1*, 1755–1762. [CrossRef]
28. Calderón, A.; Boubeta-Puig, J.; Ruiz, M. MEdit4CEP-Gam: A model-driven approach for user-friendly gamification design, monitoring and code generation in CEP-based systems. *Inf. Softw. Technol.* **2018**, *95*, 238–264. . [CrossRef]
29. Aramini, U.; Lanzi, P.; Loiacono, D. An Integrated Framework for AI Assisted Level Design in 2D Platformers. In Proceedings of the 2018 IEEE Games, Entertainment, Media Conference (GEM), Galway, Ireland, 15–17 August 2018.
30. Bontchev, B.; Panayotova, R. Towards automatic generation of serious maze games for education. *Serdica J. Comput.* **2018**, *11*, 249–278. [CrossRef]

31. Bontchev, B.; Vassileva, D.; Dankov, Y. The APOGEE software platform for construction of rich maze video games for education. In Proceedings of the 14th International Conference on Software Technologies (ICSOFT 2019), Prague, Czech Republic, 26–28 July 2019; Volume 1, pp. 491–498. [CrossRef]
32. Grammatikopoulou, A.; Laraba, S.; Sahbenderoglu, O.; Dimitropoulos, K.; Douka, S.; Grammalidis, N. An adaptive framework for the creation of exergames for intangible cultural heritage (ICH) education. *J. Comput. Educ.* **2019**, *6*, 417–450. [CrossRef]
33. Contreras-Espinosa, R.S.; Eguia-Gomez, J.L. A Prendizaje Informal Con Mods Para Videojuegos. *Educ. Humanidades-Digit.-Aprendiz. Tecnol. Cibercultura* **2019**, *1*, 301–319. [CrossRef]
34. Hauck, E.; Aranha, C. Automatic Generation of Super Mario Levels via Graph Grammars. In Proceedings of the IEEE Conference on Computatonal Intelligence and Games, Osaka, Japan, 24–27 August 2020; pp. 297–304. . [CrossRef]
35. Marín-Vega, H.; Alor-Hernández, G.; Colombo-Mendoza, L.O.; Bustos-López, M.; Zataraín-Cabada, R. *ZeusAR: A Process and an Architecture to Automate the Development of Augmented Reality Serious Games*; Springer: New York, NY, USA, 2022; Volume 81, pp. 2901–2935. . [CrossRef]
36. Kitchenham, B.; Linkman, S.; Law, D. DESMET: A methodology for evaluating software engineering methods and tools. *Comput. Control. Eng. J.* **1997**, *8*, 120–126. [CrossRef]
37. Quinn, M.M.; Symborski, C.; Barton, M.; Korris, J.; Falstad, T.; Granato, S. Methodology for playtesting serious games: A case study using a mixed method approach. In Proceedings of the IEEE Consumer Electronics Society's International Games Innovations Conference, Kyoto, Japan, 22–26 July 2013; pp. 222–227. [CrossRef]
38. Fu, F.L.; Su, R.C.; Yu, S.C. EGameFlow: A scale to measure learners' enjoyment of e-learning games. *Comput. Educ.* **2009**, *52*, 101–112. [CrossRef]
39. Phan, M.H.; Keebler, J.R.; Chaparro, B.S. The development and validation of the game user experience satisfaction scale (GUESS). *Hum. Factors* **2016**, *58*, 1217–1247. [CrossRef]
40. Player Experience in Educational Video Games. Using Playability Heuristics to Evaluate Player Experience in Educational Video Games. *J. Theor. Appl. Inf. Technol.* **2020**, *98*, 3632–3642.
41. Law, E.L.C.; Brühlmann, F.; Mekler, E.D. Systematic review and validation of the game experience questionnaire (geq)-implications for citation and reporting practice. In Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play, Melbourne, VIC, Australia, 28–31 October 2018; pp. 257–270.
42. Zubrow, D. Software quality requirements and evaluation, the ISO 25000 series. In *Software Engineering*; Software Engineering Institute: Pittsburgh, PA, USA, 2004.
43. Roa, P.A.; Morales, C.; Gutiérrez, P. Norma iso/iec 25000. *Tecnol. Investig. Acad.* **2015**, *3*, 27–33.
44. Symborski, C.; Barton, M.; Quinn, M.M.; Korris, J.H.; Kassam, K.S.; Morewedge, C.K. The design and development of serious games using iterative evaluation. *Games Cult.* **2017**, *12*, 252–268. [CrossRef]