Multimedia Tools and Applications (2022) 81:2901–2935
https://doi.org/10.1007/s11042-021-11695-1

ZeusAR: a process and an architecture to automate the development of augmented reality serious games

Humberto Marín-Vega[1] · Giner Alor-Hernández[2] · Luis Omar Colombo-Mendoza[3] · Maritza Bustos-López[4] · Ramón Zataraín-Cabada[5]

Received: 22 December 2019 / Revised: 10 February 2021 / Accepted: 25 October 2021 /
Published online: 9 November 2021

Abstract

A software development process comprises all the steps necessary to produce a software product. This research proposes a new process for developing augmented reality serious games (ARSGs), which comprises three phases: analysis, configuration, and generation. The analysis phase involves examining the standard project structure of a typical serious game to identify the applicable actions to which AR content can be added. The configuration phase involves configuring the AR features and the AR library to be integrated into the game structure. Finally, the generation phase entails inserting the AR code in the game structure, as well as all the files necessary to in-corporate AR features in the game. Our process allows generating ARSGs step by step, both easily and rapidly. To implement this process, we propose an ARSG development software architecture. The underlying assumption of this architecture is that software development tools are essential validation elements of the software generation process. By implementing our ARSG development process through a software architecture, we provide users with a formal, automated method for creating ARSGs. The ARSG development process and the architecture, collectively called ZeusAR, are not dependent on specific software development technologies and/or programming languages. To perform a proof-of-concept of our process, we developed an ARSG generator tool based on the proposed architecture and conducted a case study in which geometry ARSGs are developed to help high school students learn about geometric shapes and their properties. We conducted a qualitative evaluation of the ZeusAR tool through a System-Usability-Scale (SUS) based survey, which was administered to a group of geometry teachers to evaluate the tool's characteristics, such as ease of use and ease of configuration. Additionally, a group of software developers and professors assessed the performance of the ZeusAR tool in terms of game development time.

Keywords Augmented reality serious game · Software process

✉ Giner Alor-Hernández
galorh@orizaba.tecnm.mx

Extended author information available on the last page of the article

1 Introduction

Education can be enhanced with new technologies and methodologies that both support the teaching-learning process and contribute to the generation of high-quality educational content. However, these tools can satisfactorily fulfill their goals only if we consider the advancements in technology that have had a positive impact on educational processes during the last decades. Moreover, if compared to a few decades ago, students now interact differently with educational content, thus meaning that they learn and are entertained differently. Nowadays, there is a need to shift from traditional teaching models to active, learner-oriented forms of learning [7, 39].

Rule-based serious games provide games with a context in terms of challenges, goals, and tasks. In games, rules are constraints limiting the actions of players [10]. Serious games help players build content knowledge and improve skills and competencies in order to achieve learning outcomes. Rule-based games may have an either emergence structure or a progression structure, yet in all cases they comprise rules and challenges for learners. According to Juul [20], emergence games are structured by a small number of rules combined with a large number of variations, for which players must design strategies to handle. Examples of emergence games include strategy, action, and board games. On the other hand, in progression games players must perform a predefined set of actions to complete the game. Also, in progression games, game designers have control over the sequence of events, even though some rule games may be influenced or changed by the players' actions [10].

Serious games provide educational content and learning experiences. However, when Augmented Reality (AR) technologies are introduced, serious games ensure meaningful learning opportunities where students connect theoretical knowledge with experience. Augmented reality serious games (ARSGs) focus on two aspects that are equally important for knowledge acquisition and building: motivation and interaction [50]. Higher learning rates and skill improvement can be expected from ARSGs if compared with traditional learning and training methods. Consequently, ARSGs have the potential to increase learner engagement, interest, and motivation [11]. Moreover, game information is more condensed, thus more easily assimilated than a traditional serious game. In other words, AR technologies introduce content in such a way as to transform the learning environment into an exciting and engaging place. From this perspective, interaction is a key element of serious games, because it allows them to combine game activities with the real environment [29].

AR is an important technological resource in education, and AR educational applications are as popular in the classroom as they are outside of it. They are widely used for in-class projects or as complementary activities, with students and teachers being the creators of new knowledge [34]. As its main advantage, AR combines digital information and physical information through different technological supports, such as tablets and smartphones, to generate new enriched training scenarios [5]. Additionally, AR systems and environments help learners to develop new skills and build new knowledge more effectively, if compared to other technology-enhanced learning environments [14]. AR as a concept, rather than a type of technology, is more productive for educators, researchers, and designers, since it creates new challenges for educators [48].

In their work, Rosenbaum [38] found that AR learning environments promoted learner understanding of dynamic models and increased motivation and interest, which in turn helped learners develop better research skills and gain more accurate knowledge of the topic being taught [44]. Based on the prevalence of information delivery uses of AR in education, it is claimed that the merit of having students design AR experiences in order to develop their higher-order thinking capabilities [8] promotes enhanced learning [1]. Additionally, gamification and role-play-based AR have been applied to enhance motivation and a sense of authenticity [38], while AR mobile games allow learners to better organize, search, and evaluate data and information [24].

Any software development process comprises the necessary steps to produce a software product. Understanding software product requirements, including regulatory requirements, is an essential part of the software development process. However, in the case of serious games, specific processes or formal guidelines for their development are lacking, especially when emerging and enriching technologies, such as AR, are used. In this work, we propose a software development process that relies on AR to enrich serious games and allows users to build attractive knowledge generation experiences. It is important to define how AR features can be integrated into non-AR serious games to propose a standard AR implementation process that can be followed irrespective of the programming technology or language being used. In this sense, our work aims at proposing a standardized process for incorporating AR characteristics in serious games. By following this process, users can either develop AR serious games from scratch or incorporate AR characteristics into existing ones.

As a major benefit, a standardized process for incorporating AR characteristics in serious games can help developers decide which game actions are applicable to the games to be developed. Also, a standardized process would allow them to better manage and implement game resources, which has the potential to increase the satisfaction of learners. The interest of many researchers in finding a standardized procedure for ARSG development arises from the fact that software development processes can be difficult. Developers are often challenged to comply with a large number of requirements (including regulatory requirements). Moreover, software products can be particularly complex, especially when techniques such as AR are used.

A process is a series of successive steps necessary to achieve a specific goal. Defining the software development process, whether for serious games or for educational applications, allows developers to set up the functions that are necessary in the software to be developed. In this work, we propose an ARSG development process that relies on AR content and game attributes. As a test case, we develop an ARSG generating tool that uses a typical serious game and incorporates AR characteristics in it. Our proposed process comprises three stages: analysis, configuration, and game generation. The analysis stage involves determining whether a non-ARSG has the potential to become an ARSG. Then, the configuration stage involves configuring the applicable actions of the non-ARSG to which AR content can be added. Finally, at the game generation stage, the ARSG is developed. Our ARSG generating tool has a layered architecture to ensure easy interaction between its components.

The goal of this work is to introduce an ARSG development process that can be used in multiple contexts. Similarly, we propose a software architecture for developing an ARSG development generator; our approach as a whole is called ZeusAR. Both the process and the architecture can be used in multiple contexts and are intended to guide users, step by step, in the development of ARSGs, regardless of their experience in software development and/or knowledge of serious games, video game engines, and/or AR libraries. This work is an extension of a previous work, in which we developed Zeus as a tool for generating rule-based serious games with gamification techniques [31]. The major difference between Zeus and the ARSG generator tool herein developed based on the proposed architecture, is that Zeus adds gamification techniques into serious games, whereas the ZeusAR tool adds AR features into existing serious games – including those generated with Zeus. Additionally, to meet its goal the ZeusAR tool first identifies the actions of a serious game to which AR features can be added. Then, the tool directly inserts AR code to the pre-existing serious game code to successfully add the AR features.

We validate our work with two independent methods. First, we perform a qualitative evaluation of ZeusAR, our ARSG generator tool. To this end, we asked a group of teachers to develop geometry ARSGs for secondary education students. Then, the participants were asked to evaluate the usability of ZeusAR using a System Usability Scale (SUS)-based survey. Second, we conducted a performance analysis of the tool in terms of how much time it takes both experienced and non-experienced users to develop an ARSG.

2 State of the art

Many initiatives are constantly developed in a variety of contexts to improve the process of developing educational games, especially by emphasizing on the use of augmented reality serious games. This section discusses relevant works on and initiatives in ARSG development. The literature review is divided into two categories: 1) methodologies and models for ARSG development and 2) tools and architectures for ARSG development. Additionally, the review takes as basis some of the guidelines proposed by Kitchenham [21] for performing systematic reviews, namely those guidelines for selecting data sources. First, we determined the inclusion criteria to set the boundaries for the review. The inclusion criteria included digital versions of case studies, state of the art reviews, and original research papers on AR and Mixed Reality (MR), all published in English in both journals and conference proceedings and found online through a keyword-based search. On the other hand, we avoided thesis repository searches and discarded data sources written in languages other than English.

The search for primary studies was chiefly conducted using web search engines provided by digital research libraries on the Internet. After conducting exploratory searches in candidate libraries using keywords and synonyms extracted from our research questions, we decided to use the following search string: *(Augmented Reality OR Mixed Reality) AND (tool OR platform OR development process OR development method) AND (serious game OR educational game OR gamification technique)*. Finally, the data sources were extracted from seven digital libraries: ScienceDirect, IEEE Xplore Digital Library, SpringerLink, ACM Digital Library, Wiley Online Library, Taylor & Francis Online, and IOS Press Content Library.

2.1 Methodologies and models for AR serious game development

In [28], the author sought to design and implement user-friendly generic pervasive interfaces that could be used by a wide range of users, including people with disabilities. The generated interfaces allowed seamlessly interaction between the users and the superimposed environmental information. In [27], the authors examined some design and implementation issues of serious games that make use of tangible AR environments. The authors decided to determine whether ARSGs can be used effectively in the real gaming world. To demonstrate this, the authors evaluated two ARSGs, ARPuzzle (a puzzle game) and ARBreakout (an arcade game) and concluded that ARPuzzle was a more effective educational game than ARBreakout. According to the authors, ARPuzzle increased collaboration and promoted feedback. In [27, 28], the authors studied ARSGs and evaluated their implementation on mobile devices. In this work, we propose a standardized process for developing ARSGs.

In [41], the authors proposed an Interactive Educational Content (IEC) for secondary education that uses AR and 3D visualization technologies. The authors discussed IEC's module interaction algorithm and chose Unity, a multiplatform system real-time development of 2D and 3D applications, to develop the IEC. Also, the IEC was based on real physical and mathematical models. 3D visualization is important when presenting AR content; hence, ZeusAR includes 3D models but implements other frameworks, such as AR.JS, to incorporate AR characteristics in the serious games. In [16], the authors introduced a pattern-based framework for sensor-based AR games. The authors highlighted approaches toward specifying design patterns, created a framework of design patterns for AR games, provided ideas toward the construction of games based on these patterns, and exemplarily adapted a sample of them for Microsoft's Hololens using the Unity game engine. On the other hand, ZeusAR relies on JavaScript frameworks Ar.JS, Awe.Js, and Three.Js to incorporate AR characteristics in the games. We selected these technologies for the proof of concept due to the growing preference of HTML5/JavaScript-based games over games based on proprietary technologies, such as Unity.

In [35], the authors proposed a location-based model designed to be integrated into other serious game authoring tools supporting other game genres. This model is based on standards and aims at being compatible with existing map models, but it also creates a layer of inter-action especially valuable in location-based serious games.

In [3], the authors introduced a project aimed at improving attention in ADHD children using a case study that helps evaluate the effectiveness of ARSGs. To this end, the authors developed a serious game named ATHYNOS using a methodical formal game design approach. The participants who played ATHYNOS during various sessions improved significantly in their daily-life functioning across domains of time management and social skills. As a result, the authors reported an improvement in the participants' level of concentration. Note that ATHYNOS was developed to assess the effectiveness of ARSGs, whereas our proposal, ZeusAR, seeks to assess learning experiences. Even though our assessment criteria differ from those used for ATHYNOS, it is always important to assess the results obtained after the implementation of ARSGs.

In [33], the authors presented a high-quality tajweed learning tool using gamification and AR in the design and development process of the tool to make tajweed learning more captivating and meaningful. Additionally, the ADDIE model was employed to conduct the design and development processes. In [49], the authors created an AR geometry educational tool by using de-sign-based learning and game-based learning. The application used the iterative soft-ware development lifecycle that requires a series of prototypes to find the best AR game mechanics in the geometry learning process. Also, the tool implemented an AR system with object modelling approaches and a discovery-based learning approach in-to a game using marker-based AR. The work of [13] reports the use of AR technology to create visual aids through display for early childhood learning. The proposed methodology works on the principle of augmenting 3D virtual objects over the English alphabets that are used as printed markers. The important steps of a typical marker-based AR application are (i) detect markers in the field of view (FOV) of the camera, (ii) identify the marker, (iii) estimate the pose of the marker, and (iv) render 3D virtual content over the marker in a live video stream. We have formulated the marker identification process as a classification problem which has been solved with the help of convolutional neural networks (CNN). What can be concluded from this paragraph is that the works of [13, 33, 49] propose AR learning tools, whereas

ZeusAR proposes an ARSG generation tool where users can customize the topic to be learnt through the ARSG.

In [23], the author reviewed existing practices in AR teaching across 17 courses delivered by leading universities in order to understand current trends in AR education. The author analysed five categories: prerequisites, curriculum objectives, learning outcomes, teaching methods, and assessment means. In the end, the author found that some courses use the non-traditional method of teaching – “work-shop- type class facilitation method” and the use of a variety of assessment methods. As the authors claim, it is important to identify the teaching method so as not to deviate from the learning objective when implementing AR. To prevent AR from becoming a distraction for users, ZeusAR ensures that learning outcomes are achieved through learning tasks.

2.2 Tools and architectures for ARSG development

The work of [18] introduced MARGE, a mobile AR game engine for developing stationary maker-driven, image-based AR games. MARGE has a two-level structure and supports the Apple iPhone, Google Android, and Symbian operating systems. Additionally, MARGE’s AR library support is not limited to the AR-Toolkit, since different AR libraries can be loaded as well. If compared to MARGE, the ZeusAR tool is not a game engine, yet it implements HTML5 game engines and JavaScript-based AR libraries to generate ARSGs. Nevertheless, the ZeusAR tool’s software architecture is independent from any implementation technology; hence, it can also be used to create ARSG generator tools based on proprietary game engines.

In [15], the authors relied on AR and gamification techniques to create an educational AR book. In this project, users interacted with a virtual laboratory and were able to perform experiments and complete challenges through gaming so as to expand and test their knowledge. By using AR and gamification techniques, the goal was to deliver a more comprehensive understanding of the subject matter while simultaneously engaging learners and increasing their enjoyment during the learning process. Indeed, the AR-gamification synergy is an effective combination to reach learning outcomes; hence, we also make use of AR and gamification techniques. In fact, gamification is implemented though game attributes and learning tasks, while AR content is added to such learning tasks.

The work of [32] proposed a software architecture based on AR techniques and gestures, which acts as a “backbone” for developing wearable and mobile interaction techniques for elderly players. The gesture-based sensing system is implemented on AR glasses and supports the adaptability of the developed interaction technique to the special abilities of each individual user. Similarly, the architecture has an adaptability element that allows customizing gesture calibration. As [32] point out, defining the right software architecture is important to define the correct structure of the software product; hence, in this work we propose an architecture for the ZeusAR tool to ensure that AR features can be successfully incorporated in the serious games.

In [47], the authors presented MAGIS (Mobile Augmented-Reality Games for Instructional Support), a framework for developing educational AR games. MAGIS supports off-the-shelf, state-of-the-art technologies that enable AR tracking and rendering on consumer-level mobile devices and integrates these technologies with content generation tools that simplify educational AR game development. As regards its structure, MAGIS comprises several modular subsystems, and each of these subsystems is intended to be pluggable and/or replaceable, depending on the demands of the particular application. From a slightly different perspective, [37] proposed a framework for explaining the drivers of attitudinal and intentional reactions, such as continuance in gaming or willingness to invest money in in-app purchases. A survey among 642 Pokémon Go players provided insights into the psychological drivers of mobile AR games. Also, the results indicated that hedonic, emotional, and social benefits and social norms drive consumer reactions, while physical risks (but not data privacy risks) hinder consumer reactions. The authors also pointed out that the importance of these drivers differed depending on the form of user behaviour.

In [43], the authors introduced an optical-fog assisted EEG-based virtual reality framework, which uses the resources of the optical network to enhance the e-learning experience. The proposed framework uses an Edge-Fog layer and an Optical-Fog layer of the optical network for providing effective solutions to run EEG-based gaming applications. The framework is adaptive in the sense that the games with immediate feedback are able to respond to student experience, knowledge, and their thoughts to optimize student learning experience and improve the efficiency of the gaming applications. In conclusion, all these works propose mobile game development frameworks for different contexts, whereas we propose a software architecture in which AR libraries and video game engines represent key components. This architecture can be implemented using different AR libraries and video game engines, including proprietary ones.

The work of [2] proposed CREANDO, a platform seeking to optimize the development of gaming experiences based on pervasive narrative. Specifically, CREANDO allows for the generation of experiences in closed spaces, since these are those found in higher education institutions. The platform was validated through a test case, supported by statistical tools prior to and following the use of the experience by students. The attributes of the gameplay were evaluated, and consequently, the authors found a relationship between the pervasive games and the motivational attributes generated by the developed prototype. Additionally, the evaluation metrics concluded that pervasive games encourage student motivation, socialization, and immersion in the teaching-learning process. In [17], the authors developed a new affordable interactive multi-marker AR tool for constructing free-form modular surfaces implemented by integrating common accessible devices. The tool has two digital cameras, a head-mounted display, a processor, and two markers that enable the user to virtually see the accurate location of any proposed object in the real world. A controlling subsystem was also designed to enhance the accuracy of construction. As can be observed, all these works propose AR application platforms based on learning experiences and implement markers. ZeusAR adopts similar approaches, yet experiences are developed from learning activities, present in game attributes. Such experiences justify the learning activities being performed. Moreover, we incorporate AR in such learning activities. Such key points are known as applicable actions.

In [19], the authors proposed a set of design guidelines and recommendations for implementing AR intelligent tutoring systems. These include (1) A generalized modular architecture to support a variety of domains and pedagogical approaches; (2) Improved authoring capabilities for authoring AR courses; (3) Sparing use of annotations to avoid overwhelming the learner; (4) Support for networking and an abstraction library for sharing data between the intelligent tutoring system and AR. Conversely, the work of [40] introduced a software architecture for large-scale distributed AR services and experimental evaluation of the main component – Semantic Augmented Reality Middleware. The architecture is based on a client-server design, which supports semantic modeling and building contextual AR presentations for a large number of users. The architecture is based on the Service Oriented Architecture (SOA) paradigm that enables building distributed systems that provide application functionality as services to either end-user applications or other services. Defining a software architecture is important, since it is the point of reference for developing software. In this work, we propose a software architecture for ARSG development, which is the result of our ARSG development process.

In sections 2.1 and 2.2, we discuss a range of methodologies, models, architectures, and tools for ARSG development. As can be observed from Table 1, the main goal of most of these initiatives is to design, develop, or implement serious games with AR characteristics that can be easily played by all kinds of users, including students, people with disabilities, and autonomous learners seeking to gain knowledge in a particular area.

In most of the reviewed serious games, competence development is achieved by encouraging and reinforcing skills such as comprehension of abstract concepts and operations, explanation of topics, and argumentation, among others. As demonstrated in the works listed in Table 1, ARSGs are an opportunity to build knowledge, since they become auxiliary tools in the teaching-learning process. In the reviewed works, we found that authors paid close attention to aspects such as software design and implementation within environments that facilitate knowledge acquisition and testing. From this perspective, we conclude that ARSGs increase student engagement during the learning process, and such a process becomes more meaningful thanks to the experiences and competences developed as a result of playing ARSGs.

Table 1 summarizes our review of state-of-the-art ARSG development initiatives. In their work, McCallum et al. [43] introduced an architecture for ARSG development that is based on AR and gestures. Similarly, works such as those of [2, 17, 33, 35, 49] propose ARSG development tools with important characteristics, such as user experience and interaction, AR framework implementation, and/or dependency on a pre-existing AR framework, such as Unity [16, 41, 43]. The difference between such initiatives and our work lies in the fact that we propose a software architecture for developing an ARSG generator tool. Moreover, this architecture can be used in multiple contexts, and it is intended to guide users step by step throughout the entire ARSGs development process, regardless of users' experience in software development and / or knowledge of serious games, video game engines, and / or AR libraries.

3 The ARSG development process

To automatically generate ARSGs, it is important to design a standard process to be used as the basis for generating ARSGs with different game attributes. Our ARSG development process comprises three stages: analysis, configuration, and generation. Each of these stages comprises a series of steps, described below:

Analysis stage This stage involves reading the base game structure to identify applicable actions in which AR characteristics can be introduced.

As depicted in Fig. 1, the analysis stage of our ARSG development process com-prises the following steps:

1. **Read project data:** The zipped file introduced by the user is read, which contains the video game base project in HTML5 or proprietary technologies.
2. **Validate file and correct structure.** It is verified that the zipped file has the correct structure of the folders and files making up the base project, depending on the game selected.

Table 1 Analysis of methodologies, models, architectures, and tools for ARSG development

Author	Use of augmented reality	Framework for AR	Tools for ARSG development	Architectures for ARSG development	Methodologies for ARSG development	Models for ARSG development	Serious game development
McCallum et al. [43]	Yes, AR glasses	No	Yes	Yes, an architecture based on AR and gestures	No	No	Yes, AR games
Sannikov et al. [41]	Yes, combination of AR and 3D technologies.	Yes, Unity, Vuforia, and OpenCV.	No	No	No	Yes, real physical and mathematical models in AR.	No
Emmerich et al. [16]	Yes, design patterns for AR games.	Yes, a framework for sensor-based AR games.	No	No	Yes, design patterns for AR games	Yes, patterns using Unity	Yes, AR games
Pérez-Colado et al. [35]	No	No	Yes, serious game authoring tools supporting other game genres.	No	Yes, methodology to integrate models on AR tools.	Yes, a location-based model designed to be integrated with serious games and other games.	Yes, location-based serious games.
Nurtihah et al. [33]	No	No	Yes, a high-quality tajweed learning tool.	No	Yes, a design and development process to produce a tool.	Yes, a model to conduct ARSG design and development processes.	Yes, serious games
Julio Cristian et al. [49]	Yes, AR for geometry	No	Yes, an AR geometry learning tool	No	No	Yes, a model with AR game mechanics in the geometry learning process.	Yes, game-based learning apps.
Dash et al. [13]	Yes, using AR technology to create visual-aids.	Yes	No	No	Yes, a 3D object methodology.	Yes, a marker identification process.	Yes

Table 1 (continued)

Author	Use of augmented reality	Framework for AR	Tools for ARSG development	Architectures for ARSG development	Methodologies for ARSG development	Models for ARSG development	Serious game development
Arango-Lopez et al. [2]	No	No	Yes, a platform based on game experiences	No	Yes, implementing a GeoPGD methodology.	No	Yes, gaming experiences
Sood et al. [43]	Yes	Yes, an Optical- Fog assisted EEG-based virtual reality framework	No	No	No	No	Yes, EEG-based gaming applications.
Fazel et al. [17]	Yes, multi-marker AR	No	Yes, an interactive multi-marker AR tool	No	No	No	No

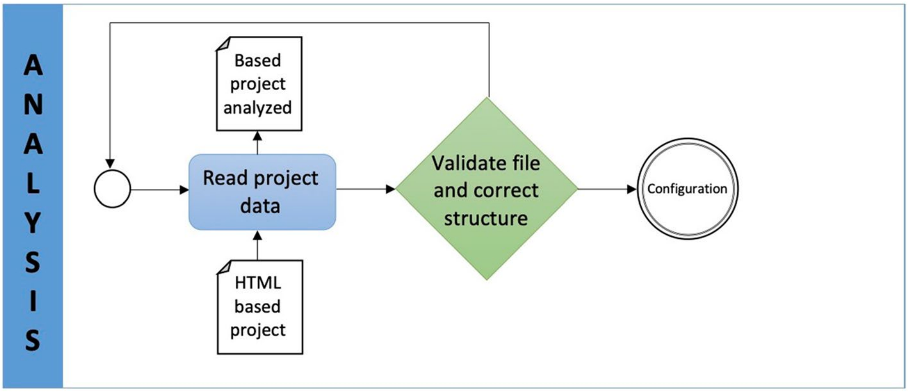


Fig. 1 ARSG generation process: Analysis stage

- If the folder structure is correct, then the first step of the configuration stage is executed.
- If the folder structure is incorrect, then return to the first step in the analysis stage.

Configuration stage This stage involves 1) customizing AR characteristics to be incorporated in the serious game, 2) selecting the AR content to be displayed, 3) selecting the library to be used to generate the functionality of the AR content, and 4) obtaining and accepting the project configuration settings.

The steps of the configuration stage, depicted in Fig. 2, are discussed below:

1. **Retrieve applicable actions.** Depending on the selected project, our ARSG development tool retrieves the applicable game actions to which AR content can be added. Each possible project corresponds to a particular game category, and the applicable

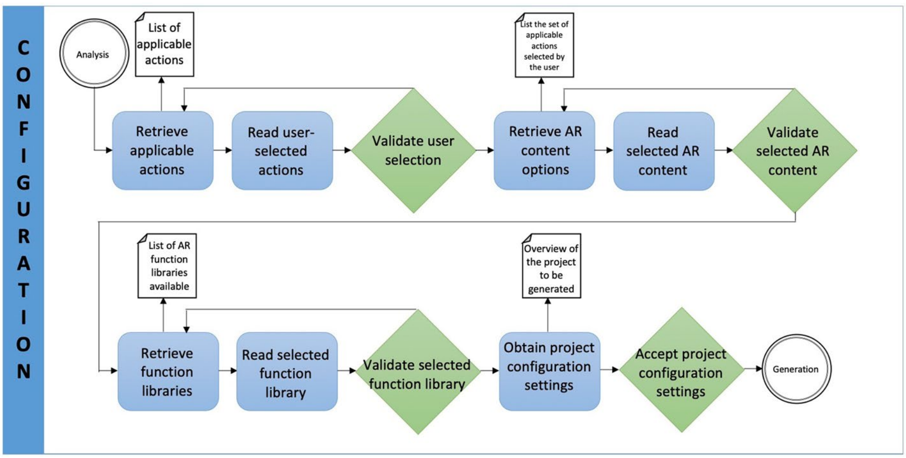


Fig. 2 ARSG generation process: Configuration stage

actions for that game category need to be catalogued. Applicable actions include, for example, the beginning and end of a game and scoring, among others. Predefined plain text configuration files are used to retrieve applicable actions in the games. These files specify the code elements from a base project in which AR code will be added, according to the game category. Consider, for instance, the game of Snakes and Ladders. The configuration file may specify the names of some JavaScript files defining the functions to add AR features into the events that are triggered when the game starts and ends, as well as when climbing a ladder or falling down a snake. As output, a list of applicable actions in which AR content can be integrated is listed. Then, the user must select at least one action in which he/she wants to incorporate AR content. The content of this sample configuration file is shown in Fig. 3.

2. **Read user-selected actions.** The applicable actions selected by the user are read and identified. Users must select at least one action to move forward to the next step.
3. **Validate user selection.** It implies validating that the applicable actions selected by the user correspond to the type of game to be developed. The system uses a compatibility tree to validate that the user has selected at least one applicable action. The concept of compatibility tree is inspired by that of dependency tree, from the software project management domain. The compatibility tree is built from a predefined XML file containing all the possible configurations that are compatible with each other. This tree is actually used throughout the entire configuration stage, in all validation steps. A simplified version of our compatibility tree, in which some nodes are shown collapsed (see nodes labeled as "..."), is depicted in Fig. 4. As can be observed, first-level nodes in this tree, which are labeled as "game", represent the types of games supported, each of which contains one child element labeled as "action" per each game action that is applicable to a game type. Each one of these second-level nodes in turn contains one child element labeled as "marker" per each type of AR content that is compatible with an applicable action. Finally, each one of these third-level nodes contains a child element labeled as "AR-settings" that in turn contains one child element labeled as "library" per each AR function library that supports one type of AR content.
4. **Retrieve AR content options.** The set of applicable actions selected by the user are obtained and listed. Then, for each of these actions, the user must select the file containing the AR content to be incorporated. AR files can be image files (PNG, JPG o JPEG), text files (.txt) or model formats (OBJ, COLLADA, PLY, GlFT).
5. **Read selected AR content.** The AR content selected by the user is read. Image files can only be in JPEG, JPG, or PNG formats.

Fig. 3 Sample plain-text configuration file

```
Game_category=rules
Game_type=snakes_and_ladders
Event_start=index.js
Event_end=index.js
Event_climb=index.js
Event_slide_down=index.js
```

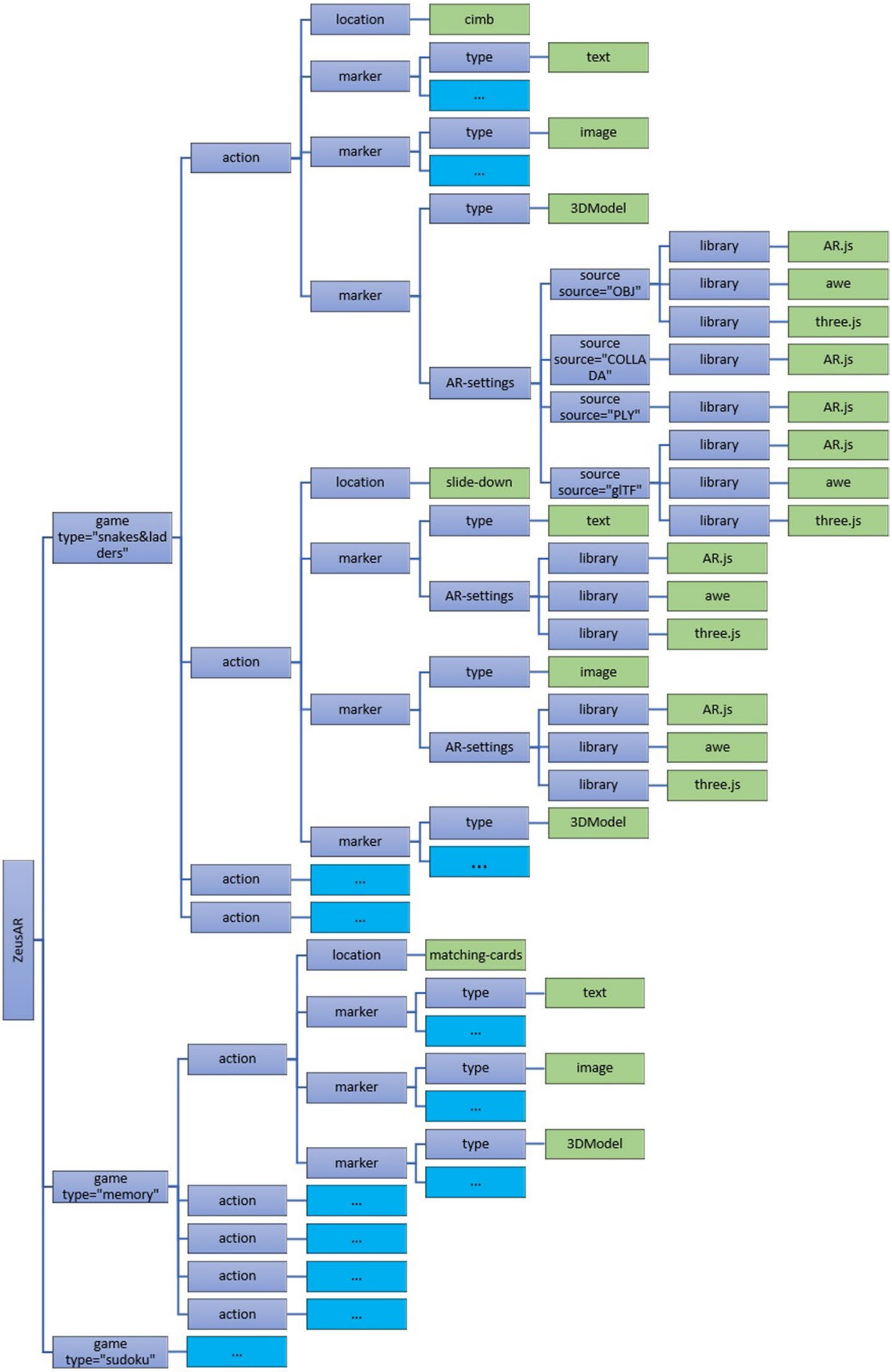


Fig. 4 Comptibility tree

6. **Validate selected AR content.** It implies validating that the AR content is applicable to the selected actions. To this end, the system relies on a compatibility tree, as previously mentioned.
7. **Retrieve AR function libraries.** The AR function libraries available are retrieved. Then, a list of them is displayed. All the applicable actions for each AR function library available need to be catalogued. The user then selects one of these libraries, depending on the game to be developed.
8. **Read selected AR function library.** The AR function library selected by the user is read.
9. **Validate selected AR function library.** This implies validating that the user selected an AR function library that is compatible with the type of AR content previously selected. To this end, the system relies on a compatibility tree, as previously mentioned.
10. **Obtain project configuration settings.** The information on the previous development steps is obtained, and an overview of the project to be generated, including information on its AR features, is displayed.
11. **Accept project configuration settings.** It consists of validating that the user accepts the project configuration settings to move to the generation stage.

Generation stage At this stage, the new structure of the ARSG project as well as the downloadable file are generated.

Figure 5 depicts the generation stage in our ARSG generation process. The steps of this stage are described as follows:

1. **Generate files and folders in the base project.** The system generates the files and folders necessary to create a new AR project from the base project. This step depends on the type of serious game and applicable actions selected by the users in previous steps.
2. **Modify applicable files.** This step involves modifying all the necessary base project files to both add AR content to the serious game and implement the selected AR function library. Mainly, labels, functions, and methods for AR content integration are injected

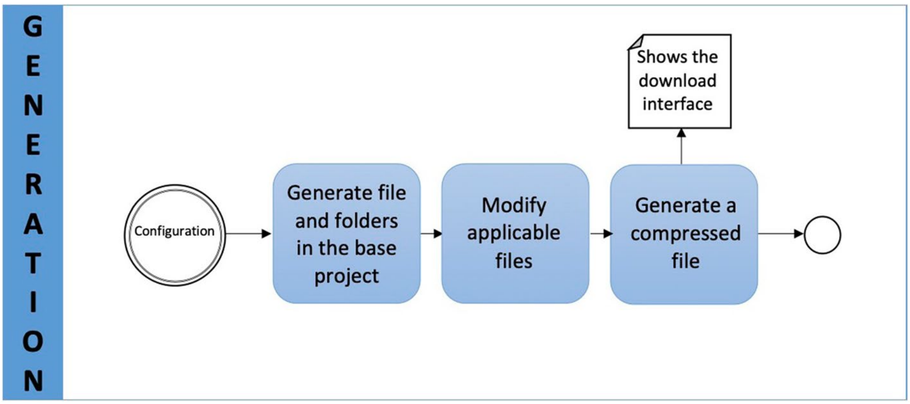


Fig. 5 ARSG generation process: Generation stage

into the base project files. To this end, the system reads the content from the script files as specified in the corresponding plain text configuration file. Script files (which could be JavaScript files) include source code; that is, the labels, functions, and methods needed to add AR features to the base project. The content of these script files is then injected into the elements of the base project code specified in the configuration file. The resulting files are finally used to replace the original files of the base project.
3. **Generate a compressed file.** A zip file of the new AR project is generated. Then, the user is given the option to download the file and the downloads interface is shown.

A development process can be defined as a formal sequence of actions that must be performed to reach an outcome. Our ARSG development process sets an orderly sequence of steps that are necessary to integrate AR characteristics into pre-existing non-AR games. To this end, our process identifies applicable actions within the non-AR games. Perhaps the main constraint of our process is the fact that it is mostly applicable to educational games. However, entertainment games may be used, as long as they comply with the necessary applicable actions.

4 An architecture for ARSG development

4.1 Design of an architecture for ARSG development

According to Bass et al. [6], the software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them. As a point of reference, we relied on the software architecture proposed by [31] to develop a generic ARSG development process that integrates AR characteristics and game attributes into pre-existing non-AR games. Our architecture has a layered design that ensures a clear organization of its components and their correct interaction. Also, these layers allow for software scalability and easy maintenance, since the tasks and responsibilities of the ARSG generator tool are evenly distributed. Figure 6 introduces a visual representation of our ARSG development architecture.

Every layer in the architecture is responsible for a particular function, described as follows:

- `Presentation layer`: Interacts with the ARSG development platform and is composed of a user interface (UI). In the presentation layer, users authenticate their identity, use the wizard to select the AR features, and download the ARSG in the generator component.
- `Application layer`: Accesses the services layer and simplifies service information, making it interface-readable data.
- `Services layer`: Provides a set of services (modules) for generating ARSGs, including (1) the serious game analyser, (2) the AR module, and (3) the builder. The serious game analyser confirms that the serious game that the user wants to generate matches the requested game structure. It also identifies the game applicable actions in which AR content can be incorporated. In turn, the AR module is responsible for adding AR content in applicable actions. Finally, the builder generates the ARSG. To this end, this

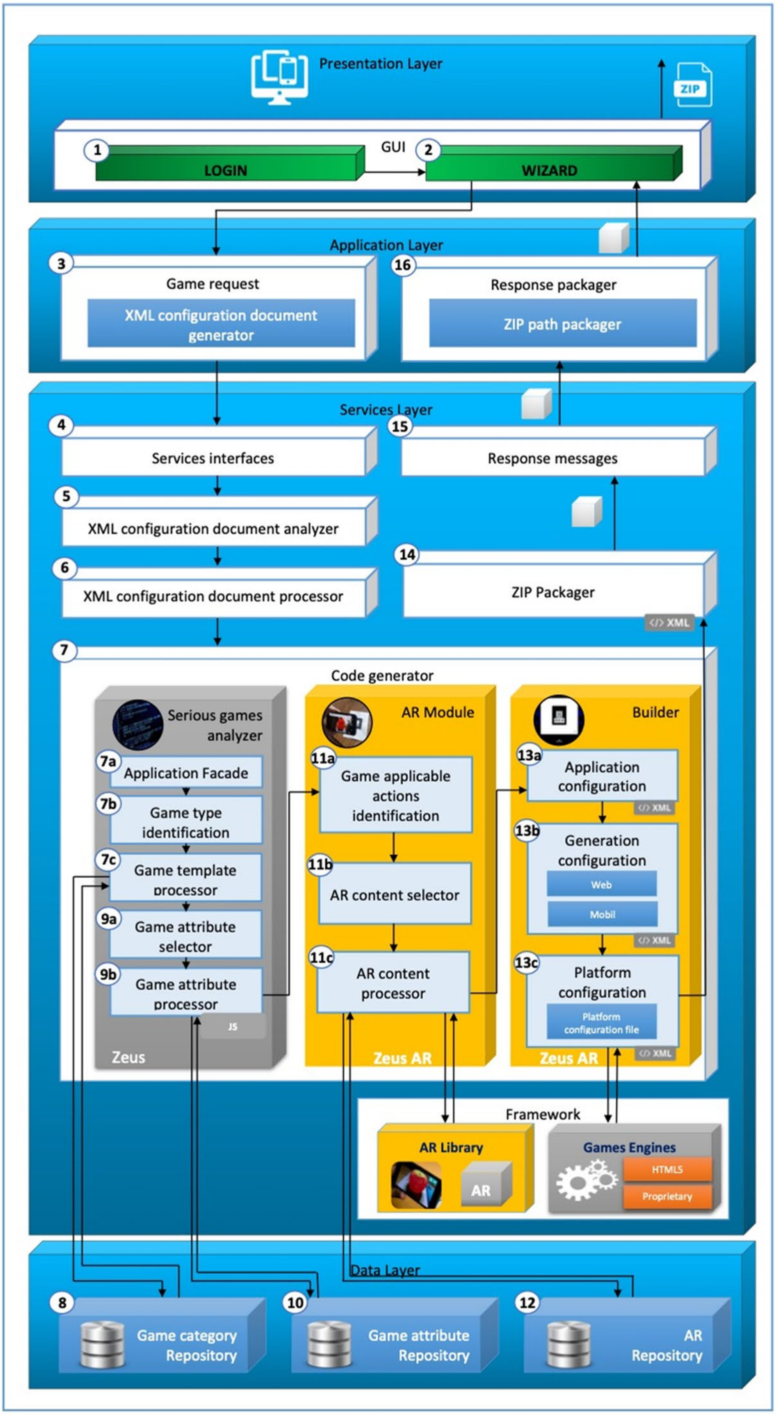


Fig. 6 An architecture for ARSG development

module relies on the AR library selected by the user. The services layer communicates directly with the data layer to access the repositories.

- `Data layer`: Stores the repositories containing the necessary data to develop the ARSG.

Similarly, every component in the architecture performs a particular function, described as follows:

- **Login.** It is a part of the graphical user interface (GUI) that performs user authentication to grant registered users access to the platform. For non- registered users, this component asks them to sign up.
- **Wizard.** It comprises a UI assistant to help users develop their gamified application.
- **Game request.** It processes user requests made in the wizard. The game request component generates an XML configuration file describing the requests made in the presentation layer.
- **XML file generator:** This component generates the XML configuration file that contains all the data necessary to generate the ARSG. Figure 7 depicts a simplified version of a tree data structure built from a sample XML configuration file in which few nodes are shown collapsed (see nodes labeled as "..."). During the configuration stage of our ARSG development process, the <game> and < AR-settings> child elements of the XML configuration file tree are particularly built by the XML file generator component. The resulting tree substructures, which are highlighted in Fig. 7, are matched against the compatibility tree (please see Fig. 4 in previous section) to ensure that the choices made by the user through the wizard regarding game actions, AR content and AR function libraries are compatible with each other. In fact, the <game> child element in the XML configuration file represents the specifications of the ARSG: type of game, game engine, game actions and AR content; the <AR-settings> child element wraps all the information that is required for the compilation of the ARSG, including the name of the selected AR function library.
- **Service Interface**. It guarantees communication between the application layer and the service layer components.
- **XML configuration file analyser:** It verifies that the structure of the XML configuration file is correct.
- **XML configuration file processor:** It processes the information described in the configuration file to identify the required data and send the information to the code generator component.
- **Code generator:** It is located in the services layer and generates the ARSG. Namely, the code generator analyses applicable actions within a base project, processes the AR content to be incorporated in such actions, and configures the serious game. Additionally, the code generator interacts directly with AR libraries and has the following sub-components:
- **Serious game analyser:** This module identifies applicable actions within the base project structure. Then, it analyses each of these actions to determine the type of AR content to be introduced. The serious game analyser has the following sub-components:

  ○ *Application facade*. It implements the facade design pattern to provide a much simpler interface to communicate with the other components of the code generator.

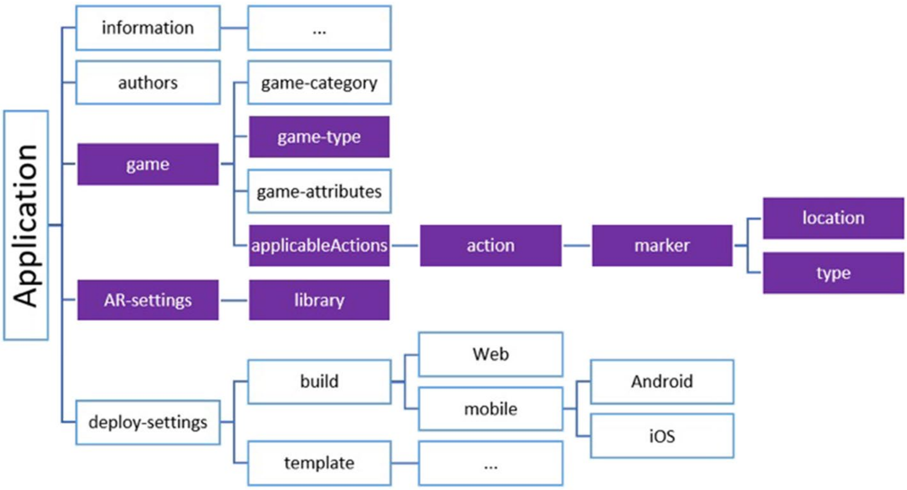


Fig. 7 Application configuration tree

○ *Game type identifier*. This sub-component analyses the XML configuration file and identifies the type of ARSG to be generated. Then, it sends this information to the game template processor.
○ *Game template processor*. It receives a message containing information on the game type to be developed. Then, it consults the game category repository to retrieve the information from the game template to identify the game attributes for that category.
○ *Game attribute selector*. This component identifies the game attributes of each game category and retrieves the attribute types accordingly.
○ *Game attribute processor*. It processes the characteristics of an attribute to generate the game attribute.

- **AR module:** It is the main sub-component of the code generator. The AR module adds AR content to the applicable actions previously identified by the serious game analyser. To this end, the AR module interacts with the AR libraries. The sub-components within this module are outlined below:

  ○ *Applicable action identifier*: This sub-component identifies the applicable actions that the serious game must contain.
  ○ *AR content selector*: The AR content to be added in the applicable actions is selected.
  ○ *AR content processor*: This processor adds AR content to a serious game through the user-selected AR library and making use of the AR repository.

- **Builder**: It builds the ARSG application. To this end, the builder configures the data of the ARSG and the building environment.

  - *Application configurator*. It gathers the necessary application development data (e.g. application name, development date, developer name) and saves them in the XML file.

  - *Generation configurator*. It gathers information on the platform selected to deploy the ARSG. ZeusAR can develop game applications for both the web and mobile devices; thus, the generation configurator comprises two sub-components: a web-based code generator and a cross-platform mobile code generator. The former describes information to generate web applications using HTML and JavaScript, whereas the latter describes information to generate AndroidTM or iOSTM game apps.
  - *Platform configurator*. This component generates an XML configuration file that includes a description of the characteristics and attributes of the application to be developed, either in a game engine of for the Web, as previously decided by the user.

- **ZIP Generator**. It compresses into a zip file the results of the code generator.
- **Response message generator.** It generates a response message after a game application is generated. If an error occurred during the generation process, the response message includes the details on that error.
- **Response packager**: It receives the message and a Zip file generated in the previous component and sends the file to the presentation layer.

5 Developing AR serious games

According to Lameras [26], rule game attributes must be scoring, moving, timer levels, progress bars, and game instructions, including victory conditions. In terms of learning activities, rule games can be either information transmission or individual activities. In this work, we took Zeus, a serious game generation platform developed by Marín et al. [31], as the base of our project. Zeus can develop three types of rule games – memory games, Sudoku, and Snakes and Ladders. Many other games, such as domino and chess, fit into the category of rule games, yet memory games, Sudoku, and Snakes and Ladders comply with the two learning characteristics previously mentioned: they transmit information and can be played through individual activities. Moreover, we believe that AR features in these games can enhance user learning experiences and satisfaction.

Memory games, Sudoku, and snakes and ladders have interaction rules that lead players to achieve a particular learning outcome. Some of these rules are concerned with the movements and actions that players must perform while interacting with the games.

In these rules, we identified applicable actions in which AR characteristics could be added. In this sense, applicable actions involve one user interaction with a given game attribute. Table 2 below lists the applicable actions identified in each type of game.

6 Evaluation of ZeusAR

In her work, Kitchenham [22] proposes a methodology for evaluating software engineering methods and tools. The methodology thoroughly describes some procedures for conducting quantitative case studies and qualitative analyses. Moreover, Kitchenham [22] describes three types of evaluation methods:

Table 2 Tasks in AR serious games

Game type	Action	Activation mechanism	AR content type
Memory	Start game	Activation button	Board on a flat surface.
	Wrong card	Automatic	Error message or image.
	Matching cards	Automatic	Right guess text or image.
	Exit game (unfinished)	Automatic	Text or image indicating that the game is not finished.
	End game (all cards matched)	Automatic	Game end text or image.
Sudoku	Start game	Activation button	Board on a flat surface.
	Exit game	Automatic	Text or image indicating that the game is not finished.
	Completed grid	Automatic	Game end text or image.
Snakes and ladders	Start game	Activation button	Board on a flat surface.
	Go up through ladder	Automatic	Action message or image of a person going up through a ladder.
	Go down through ladder	Automatic	Action message or image of a person going down through a ladder.

- `Quantitative methods`. They assume that researchers can identify some measurable property (or properties) of the software product or process that they expect to change. Quantitative evaluations can be organized in three different ways: case studies, formal experiments, and surveys.
- `Qualitative methods`. They use the term Feature Analysis to describe a qualitative evaluation. Feature Analysis is referred to as qualitative because it generally requires a subjective assessment of the relative importance of different features.
- `Hybrid methods`. They have both quantitative and qualitative elements.

In software engineering, qualitative methods are widely used to evaluate usability and user experience [22]. For instance, Perry [36] performed a qualitative evaluation of an AR mobile learning tool based on the users' learning experience. Evaluating software usability is key to identifying the course of a research project. That is, the level of usability of a software product determines user experience. For instance, when evaluating CREANDO [2], the authors found that the tool had the potential to increase student motivation in learning. Considering the importance of qualitative assessments, we performed a qualitative evaluation of the ZeusAR tool, our ARSG development tool.

The System Usability Scale (SUS) is a fast and reliable tool for measuring software usability [9, 42]. Also, the SUS is widely used to evaluate the usability of automatic code generation tools [12, 30, 45] such as the ZeusAR tool, which is mainly aimed for non-expert users. As an example, authors [30] relied on the SUS to evaluate LAGARTO, an authoring tool for building location-based mobile games with AR features. Specifically, the SUS was used to determine overall user acceptance of LAGARTO and compare the performance of programmer users with that of non-programmer users. SUS is also a reliable tool for other software products, such as mobile applications or TV applications, among others [4]. Additionally, SUS is employed in complex evaluations, such as that of a game-based learning system that uses the Microsoft Kinect sensor [46], or the evaluation performed to the Science Sports AR (SSAR) tool [25].

To evaluate the ZeusAR tool, we used SUS [9], which is similar to the Likert scale and can be used to measure the level of usability of an object, device, or application [9]. SUS is a fast and reliable way to assess usability [42], and multiple research works confirm that it yields accurate data. As a result, SUS is one of the most used methods to measure user experience [42, 45].

6.1 User study

As previously mentioned, this research work relies on a SUS-based survey to evaluate the usability of the ZeusAR tool. The evaluation was conducted by basic trigonometry teachers and facilitators, and the goal was to determine whether the ZeusAR tool could successfully and automatically generate code. Similarly, the evaluation comprised two phases. In the first phase, we administered the SUS questionnaire, whereas in the second phase, we conducted a performance analysis of the ZeusAR tool. This two-phase evaluation method takes as reference the method used to evaluate LAGARTO [30], which according to the authors, ensures a step-by-step usability evaluation, paying close attention to aspects such as the sample and evaluation times.

Table 3 SUS-based survey

Question
1. I think that I would like to use this tool to create augmented reality educational serious games.
2. I felt very confident using the tool.
3. I thought the tool was easy to use.
4. I think that I would not need assistance to be able to use the tool.
5. I did not need to learn a lot of things before I could get going with the tool.
6. I would imagine that most people would learn to use this tool very quickly.
7. I think that it is easy to configure game engines with this tool.
8. I think that the tool allows users to easily add augmented reality content in the games.
9. I think that target game platforms can be easily configured with this tool.

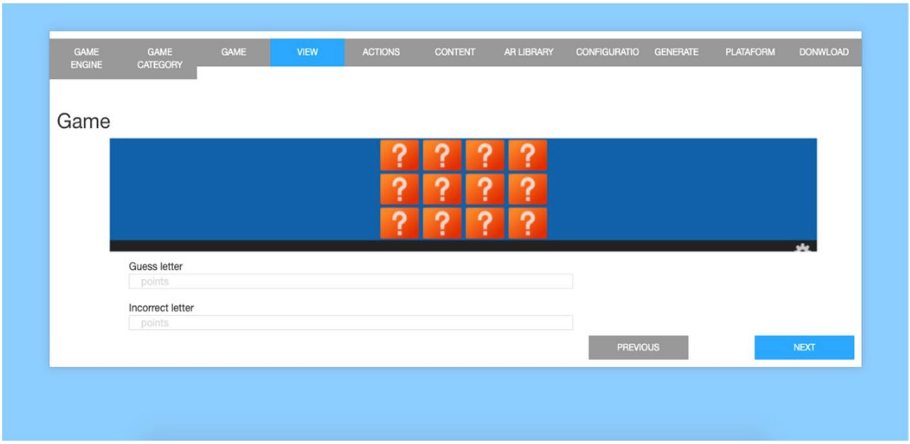


Fig. 8 Preview of the game recovered from the repository

6.1.1 Study design

ZeusAR is a tool for developing ARSGs. The development process first involves the user selecting a non-AR serious game from the tool's repository. Then, the ZeusAR tool identifies all the applicable actions in the game. Next, the user selects those actions in which he/she wishes to incorporate AR features. Finally, the user enters the configuration data of the AR game to be developed, the AR library, and the deployment platform. Perhaps the major advantage of code generation tools, such as the ZeusAR tool, is that of helping users with coding skills as well as those without coding skills to generate applications, such as game apps, easily and steadily. From this perspective, it is important to evaluate how easy it is for ZeusAR users to generate ARSGs apps with the tool. To this end, we performed a qualitative usability evaluation using a SUS-based survey.

Our SUS-based survey evaluates aspects such as ease of use, security, and user experience (Table 3). To respond to each survey item, the participants had the choice among *strongly agree*, *strongly disagree*, or another answer in between. By adapting the survey to the particular characteristics of the ZeusAR tool, we made sure to evaluate key aspects, such as how easy it is for users to configure the game engines, the AR content, and the deployment platforms, without omitting key aspects of the traditional SUS (e.g. usability and security).

After the usability evaluation, we conducted a performance analysis. To this end, we set the criteria for developing ARSGs with the ZeusAR tool. Then, we asked both programmer users (experts) and non-programmer users (teachers) to develop a game with the ZeusAR tool in the shortest possible time, thus taking game development time as the performance metric. The usability evaluation of the ZeusAR tool was conducted by 10 geometry teachers from diverse public and private high schools located in the region of Orizaba, Veracruz with at least 15 years of experience teaching math subject. The goal was to determine the average time that users spent to develop an ARSG by using ZeusAR tool. The teachers have at least 15 years of experience in teaching math subject. The teachers also participated in the performance evaluation such as non-experienced users because they have not software development experience or any type of experience related to automated development software process or source code generation process. The participants in the performance evaluation are senior programmers. All participants have experience in automated development software process or source code generation process. Some of them have experience specifically in videogames development.

6.1.2 Evaluation scenario

In this study, we asked users to develop ARSGs such as memory games, Sudoku, and Snakes and Ladders using the ZeusAR tool. These games are prestored in the tool's repository (Fig. 8). Hence, the users only had to configure the applicable actions in which AR would be introduced and to select the preferred AR library (Fig. 9). All the games developed in the evaluation were geometry games for secondary education students. The games covered topics such as triangles, rectangles, squares, polygons, and their properties (Fig. 10). Geometry is an important area of mathematics in secondary education, yet it can be both difficult to learn and little motivating. As a result, geometry teachers must find tools that help them easily design attractive teaching material, and as a result, they will be providing their students meaningful experiences for learning and building new knowledge. Finally, it is important to mention that the ZeusAR tool can generate ARSGs for various contexts.

6.1.3 Participants

The usability evaluation of the ZeusAR tool was conducted by ten geometry teachers and facilitators from diverse secondary schools. The goal of the evaluation was to confirm that the ZeusAR tool can successfully help any person (with or without coding skills, and with or without knowledge on software development) successfully and easily develop ARSGs, step by step, using a formal automatic code generation process. The evaluation took place in the computer laboratory of the Orizaba Campus of The National Technological Institute of Mexico (TecNM, by its Spanish acronym). We formed two groups of five participants (i.e. teachers and facilitators), and each group was assisted by an evaluation leader and a ZeusAR tool technician. Next, the participants were asked to develop a memory game, a Sudoku, and a Snakes and Ladders game using the ZeusAR tool. After that, the participants answered the SUS-based survey, as discussed in the study design section. To ensure that all the participants could

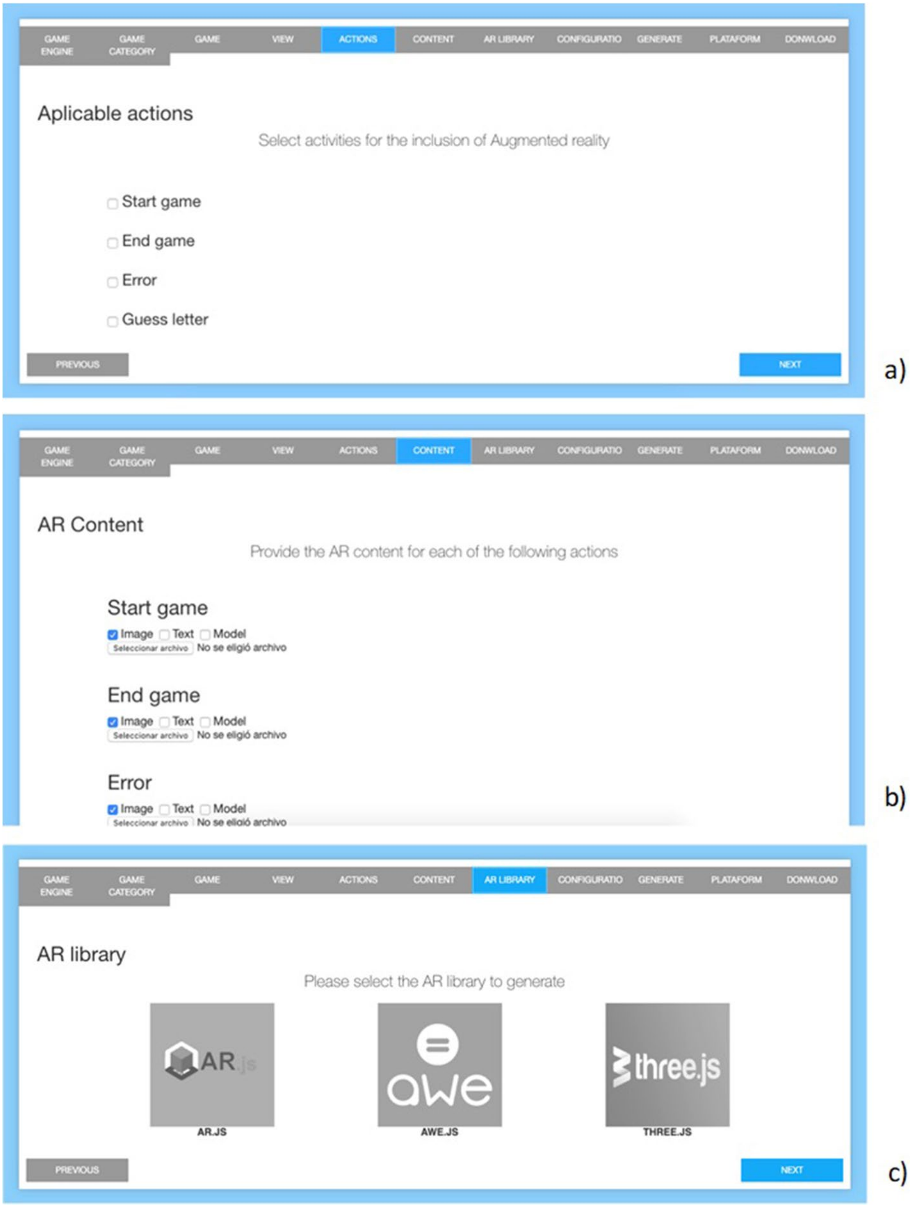


Fig. 9 AR applicable actions configuration and AR library selection: **a** AR applicable actions selection, **b** AR content configuration, **c** AR library selection

answer the survey without difficulty, we gave a brief explanation of both the original SUS and our survey.

We also conducted a performance analysis of the ZeusAR tool. To this end, we defined a sample of 20 participants: ten programmers (i.e. experts) and ten non-programmers (i.e. the teachers and facilitators involved in the usability evaluation).

Fig. 10 Game generated with augmented reality: AR content presentation in the applicable action "start game" in the game generated

The participants were gathered in the computer laboratory of the Orizaba Campus of TecNM. Once more, we formed groups of five participants. Each participant was asked to develop a serious game in the shortest possible time using the criteria listed in Table 4.

6.1.4 Data gathering procedure

During the usability evaluation, and once we briefly explained how the ZeusAR tool worked, the participants (i.e. the professors and facilitators) were given 20 min to develop an ARSG. Next, they were given five minutes to answer the SUS-based survey to evaluate the usability aspects of the tool. To ensure that all the participants could answer the survey without difficulty, we gave a brief explanation of both the original SUS and our survey. All the survey items had to be answered on a five-point rating scale, ranging from *strongly disagree* to *strongly agree*.

As regards the performance analysis, its goal was to determine the average time that it takes users to develop an ARSG with the ZeusAR tool. The performance analysis was conducted through a sample of 20 participants: ten programmers (experts) and ten non-programmers (non-experts). In both groups, we asked the participants to develop an AR memory game that complied with the specifications listed in Table 4. We used individual time counters to measure the time that it took each participant to develop an ARSG with the ZeusAR tool.

6.1.5 Data analysis and validation

All the participants successfully completed the evaluation. The results of the SUS-based survey were coded using numbers 0 to 4, the former representing a *totally disagree* response and the latter representing a *totally agree response*. The values in between represented a proportional range. As for the results of the performance analysis, we used individual time counters to measure the time that it took each participant to develop an ARSG with the ZeusAR tool.

Table 4 Specifications of the ARSGs developed in the usability evaluation

Game specifications	
Area	Geometry
Topic	Triangles, squares, polygons and their properties
Type of ARSG	Memory game, Sudoku, Snakes and Ladders
Game engine	HTML5-based
Applicable actions	AR content type
Game start	JPG image
Game end	OBJ model
Right card	Text
Wrong card	PNG image
AR library	AR.JS
Deployment platform	Web

Table 5 Results of the usability questionnaire

Question	Arithmetic mean results
1. I think that I would like to use this tool to create augmented reality educational serious games.	3.8
2. I felt very confident using the tool.	3.3
3. I thought the tool was easy to use.	3.5
4. I think that I would not need assistance to be able to use the tool.	3.2
5. I did not need to learn a lot of things before I could get going with the tool.	3.2
6. I would imagine that most people would learn to use this tool very quickly.	3.5
7. I think that it is easy to configure game engines with this tool.	3.5
8. I think that the tool allows users to easily add augmented reality content in the games.	3.5
9. I think that target game platforms can be easily configured with this tool.	3.5

6.2 Results and discussion

6.2.1 Results

This section introduces the results of the ZeusAR tool's usability assessment and performance analysis. The results summarized in Table 5 show the arithmetic mean of the survey responses. Additionally, a visual representation of these usability results is introduced in Fig. 11.

We obtained satisfactory results in the usability evaluation. In most of the survey items, the arithmetic mean is either equal to or higher than 3.5. This value is close to 4, which represents *Totally agree*. In total, 66.67% of the responses have a mean value equal to or

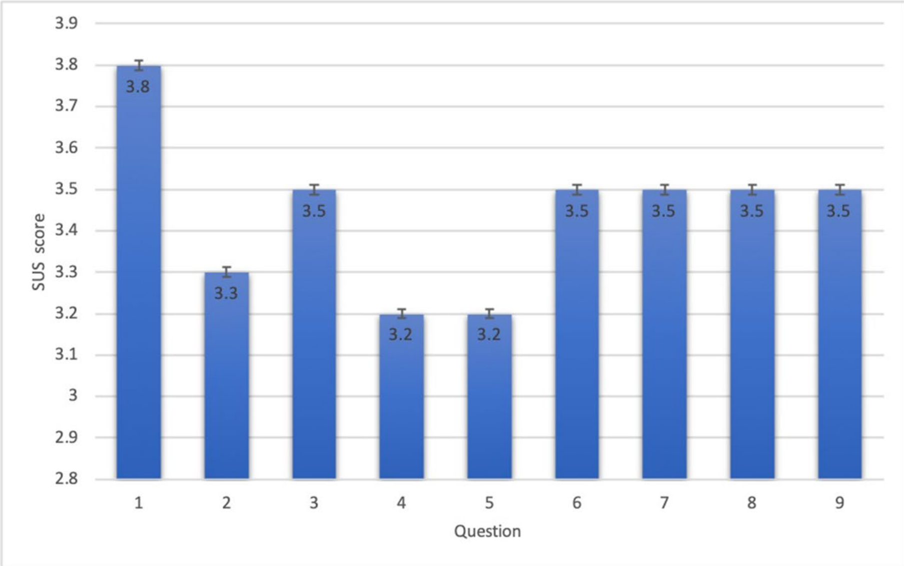


Fig. 11 Graphical representation of usability results

higher than 3.5, whereas 33% of the responses have a tendency towards 3, which represents *Agree*. In terms of user experience, the survey results showed a high median value (3.4444) with a small standard deviation, thus implying that most users agree on the fact that they had a favourable usability experience with the ZeusAR tool.

Survey questions 3 (*I thought that the tool was easy to use*) and 8 (*I think that the tool allows users to easily add augmented reality content to games*) are questions on user experience. They were formulated to assess whether the ZeusAR tool can achieve its goal, which is to develop ARSGs easily and rapidly. Specifically, 3 evaluates aspect ease of use. In this sense, note that the ZeusAR tool implements a wizard-assisted navigation system that guides users step by step along the entire AR serious game development process. The wizard in the ZeusAR tool signposts the different phases of the game development process and makes use of visual aspects, such as images and previews, to draw the attention of users and facilitate their interaction with the tool. Furthermore, users can choose the configuration of the game engine, the AR library, and the deployment platform, which simplifies the development task for non-experts. Conversely, the process of adding AR content to the games is more detailed, yet users are guided at all times. Instead of programming or modifying a configuration file, users only need to select those actions from their game in which they wish to incorporate AR content. The results in both 3 and 8 were 50% *Totally agree* and 50% *Agree*. In conclusion, we found that overall, the participants agreed on the fact it was easy for them to develop ARSGs with the ZeusAR tool.

The goal of the performance analysis was to evaluate the average time needed to develop an ARSG with the ZeusAR tool. According to the results, it takes averagely 5.51 min for programmers to develop an ARSG with the tool, whereas non-programmers take on average 8.04 min. In the first case, the standard deviation was 1.3832, whereas in the second case, the standard deviation was 1.7063. A summary of the performance analysis results is graphically introduced in Fig. 12. In conclusion, the ZeusAR tool is an interactive, easy-to-use ARSG development tool. However, notice that during the evaluation, technicians

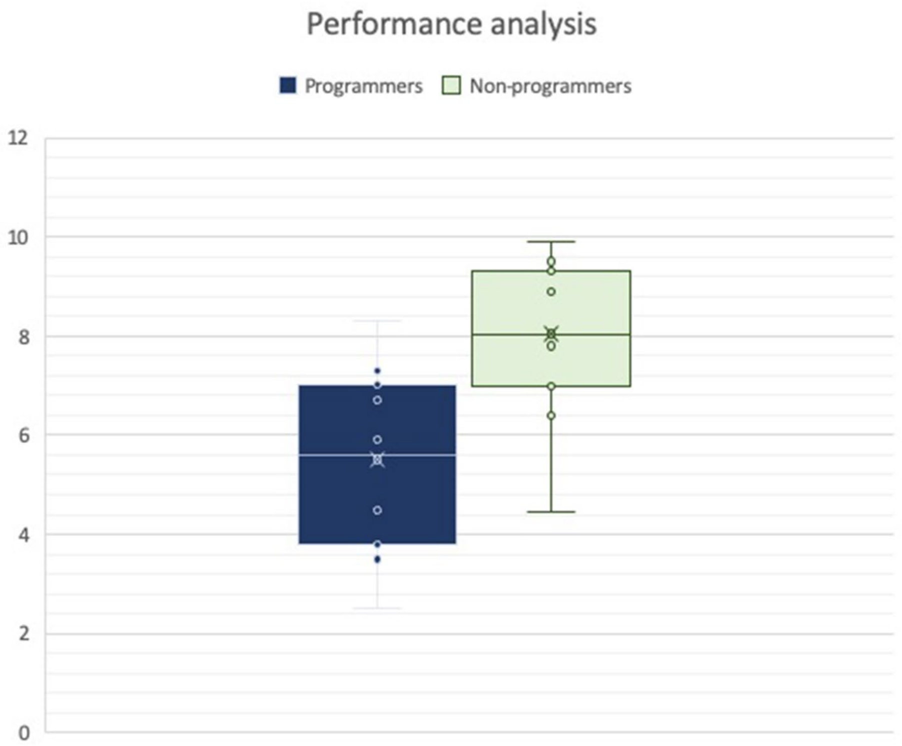


Fig. 12 Performance analysis results

guided the participants through the entire game development process. Also, for the performance analysis, we considered both the development time and testing time by the AR content reader. We did not consider the time taken to find material for the AR content.

Our results revealed that ZeusAR tool is an easy-to-use ARSG development tool. Both expert users and non-expert users can develop ARSGs in a relatively short period of time, regardless of the complexity of the games. Additionally, ZeusAR users are not required to have knowledge on software development to be able to use the tool. According to our results, minimum development time for non-expert users was 4.45 min, whereas maximum development time was 9.50 min. We found that non-expert users took more time to develop the serious games than expert-users, yet we believe this was mostly due to their age (30 to 58 years old). Moreover, not all the non-expert participants were familiar with software development tools or AR content. That said, an average development time of 8.04 min for non-expert participants is relatively short.

6.2.2 Discussion

We obtained favourable results in the usability evaluation. Most of the responses had a tendency toward *Totally agree*, which reveals that the ZeusAR tool is an easy-to-use software development tool for people without knowledge on software development or coding skills. The highest-rated aspects of the ZeusAR tool included user acceptance and the fact that it is easy to configure the game engines, the AR content, and the deployment platforms.

Table 6 T-test results

	Expert user times	Non-expert user times
Mean	5.51	8.04
Variance	3.55	2.91
Standard Error	0.59	0.54

Such results revealed that we managed to develop a software development tool that easily and safely guides users through the formal process of developing ARSGs. On the other hand, less encouraging results were obtained in aspects such as previous knowledge and the tool's technical support. These aspects showed values below the median but still fell into the *Agree* range.

The SUS yields a single value within the range 0–100 to represent a composite measure of the overall usability of a given system. To this end, the sum of the individual scores is multiplied by a factor of 2.5 to obtain a single score within the range of 0 to 100. This is due to the fact that the original SUS comprises ten questions, each contributing with a value ranging from 0 to 4 to the final usability score. In this work, however, we partially adapted the original SUS to assess the usability of the ZeusAR tool. Our adapted version of the SUS comprised nine questions; hence, we had to calculate a new factor of overall usability, namely 2.222, to compute the final usability score of the tool. After applying this new factor to the sum of the contributions of the nine scale items, we found that the overall usability score of the ZeusAR tool was 88.88. As regards the performance analysis results, we concluded that ZeusAR users can develop ARSGs rapidly. The mean value of the expert-user results was 5.51, whereas that of the non-expert-user results was 8.04. In other words, we found that, on average, users take less than 10 min to develop an ARSG with the ZeusAR tool. The gap between the mean value in the expert-user results and the mean value in the non-expert-user results is due to the age of the participants (30 to 58 years old). That is, older participants took more time in the development process than younger participants, namely up to 9.50 min. However, our results indicate that even though some users were not initially familiar with software development tools, they managed to easily and rapidly develop ARSGs with the ZeusAR tool.

Finally, we performed a t-test of two paired samples of means on the results of the performance analysis; Table 6 shows the means, variance and standard error of each one of the two samples. The goal was to prove that either the differences between the mean of expert user times and that of non-expert user times is 0, or that the means are different; to this end, we used an alpha value of 0.001.

The result proved that the mean of expert user times is actually smaller than that of non-expert user times (the null hypothesis could be rejected), $t(9) = -6.18$, $p < .001$.

7 Conclusions

AR can add virtual information to the real world, thus enriching our environment. In the education domain, AR has various advantages if compared to other technologies; for instance, it encourages more cognitive abilities. Also, games with AR characteristics yield satisfactory results, because they successfully retain player attention and promote motivation. From this perspective, ARSGs offer players the opportunity to explore new learning experiences. In this work, we proposed an ARSG development process in which AR attributes can be incorporated into the structure of traditional serious games. The underlying motivation is to offer a software development tool that easily and rapidly guides users through the entire ARSG development process. To this end, our ARSG development process is implemented in an architecture for developing educational game generators. The architecture is not dependent on specific software development technologies or programming languages. Its only goal is to guide users in the process of building game generators, thus showing users the components to be developed and how to interact with them.

Both the ARSG development process and the architecture are called ZeusAR. The goal of developing the ZeusAR tool is to verify that, if our ARSG development process is properly followed, users can rapidly and easily develop ARSGs without needing prior knowledge or skills in either programming or software development. From this perspective, the ZeusAR tool has the potential to help any teacher to create appealing support material for their courses, while simultaneously helping students build new knowledge through meaningful experiences. ZeusAR is a tool that can easily and rapidly develop ARSG using a formal AR serious game development process. Moreover, Zeus offers a safe, wizard-assisted interface that guides users throughout the entire development process. According to the usability evaluation results, most of the users who interacted with the ZeusAR tool felt comfortable with it. In terms of the tool's usability, the majority of the users either agreed or totally agreed on the fact that the ZeusAR tool is fast and easy to use, especially in terms of adding AR content in the games and configuring game engines. Finally, as regards the performance analysis results, we found that the ZeusAR tool can successfully and rapidly develop ARSGs that comply with user specifications, learning outcomes, game attributes, and learning activities.

8 Future work

As future work, we will seek to conduct an evaluation of the ARSGs developed with the ZeusAR tool in terms of learning experience and user experience (player experience) in order to measure the impact of rule-based ARSGs on knowledge building processes. Similarly, we have planned to compare the ZeusAR tool with similar tools, both reported in the literature and commercially available. Furthermore, we will seek to extend the tool's repository by adding games from the other categories proposed by Lameras, such as task-based games, challenge-based games, and goal-based games. To this end, it would be necessary first to identify all the applicable actions of each game type, and then, to build a classification matrix with the information. Finally, we will also seek to integrate other AR libraries in the repository of the proposed software architecture (e.g. Argon.JS® and A-Frame®) and support other types of AR content, such as streaming videos and GIF images.

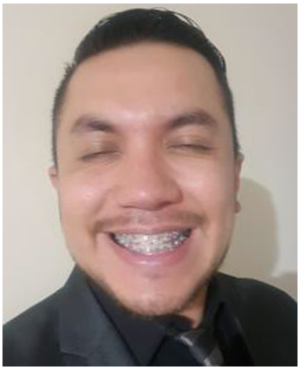
References

1. Akçayır M, Akçayır G (2017) Advantages and challenges associated with augmented reality for education: a systematic review of the literature. Educ Res Rev 20:1–11. https://doi.org/10.1016/j.edurev.2016.11.002
2. Arango-López J, Cerón-Valdivieso CC, Collazos CA, Gutiérrez-Vela FL, Moreira F (2019) CREANDO: tool for creating pervasive games to increase the learning motivation in higher education students. Telematics Inform 38:62–73. https://doi.org/10.1016/j.tele.2018.08.005

3. Avila-Pesantez D, Rivera LA, Vaca-Cardenas L, Aguayo S, Zuniga L (2018) Towards the improvement of ADHD children through augmented reality serious games: preliminary results. In: IEEE global engineering education conference, EDUCON. IEEE Computer Society, pp 843–848
4. Bangor A, Kortum P, Miller J (2009) Determining what individual SUS scores mean: adding an adjective rating scale. J Usability Stud 4:114–123
5. Barroso Osuna J, Cabero Almenara J, Moreno Fernández AM (2016) La utilización de objetos de aprendizaje en Realidad Aumentada en la enseñanza de la medicina. Innoeduca Int J Technol Educ Innov 2:77. https://doi.org/10.20548/innoeduca.2016.v2i2.2028
6. Bass L, Clements P, Kazman R (2012) Software architecture in practice, 3rd edn. Addison-Wesley Professional
7. Bekebrede G, Warmelink HJG, Mayer IS (2011) Reviewing the need for gaming in education to accommodate the net generation. Comput Educ 57:1521–1529. https://doi.org/10.1016/j.compedu.2011.02.010
8. Bower M, Howe C, McCredie N, Robinson A, Grover D (2014) Augmented reality in education - cases, places and potentials. EMI Educ Media Int 51:1–15
9. Brooke J (1996) SUS-A quick and dirty usability scale. Usability evaluation in industry. CRC Press
10. Charsky D (2010) From edutainment to serious games: a change in the use of game characteristics. Games Cult 5:177–198. https://doi.org/10.1177/1555412009354727
11. Checa D, Bustillo A (2020) A review of immersive virtual reality serious games to enhance learning and training. Multimed Tools Appl 79:5501–5527. https://doi.org/10.1007/s11042-019-08348-9
12. Colombo-Mendoza LO, Alor-Hernández G, Rodríguez-gonzález A, Valencia-garcía R (2014) MobiCloUP!: a PaaS for cloud services-based mobile applications. Autom Softw Eng 21:391–437. https://doi.org/10.1007/s10515-014-0143-5
13. Dash AK, Behera SK, Dogra DP, Roy PP (2018) Designing of marker-based augmented reality learning environment for kids using convolutional neural network architecture. Displays. https://doi.org/10.1016/j.displa.2018.10.003
14. El Sayed NAM, Zayed HH, Sharawy MI (2011) ARSC: augmented reality student card an augmented reality solution for the education field. Comput Educ 56:1045–1061. https://doi.org/10.1016/j.compedu.2010.10.019
15. Eleftheria CA, Charikleia P, Iason CG, Athanasios T, Dimitrios T (2013) An innovative augmented reality educational platform using gamification to enhance lifelong learning and cultural education. In: IISA 2013 - 4th international conference on information, Intelligence, Systems and Applications. pp 258–263
16. Emmerich F, Klemke R, Hummes T (2017) Design patterns for augmented reality learning games. In: Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics). Springer Verlag, pp 161–172
17. Fazel A, Izadi A (2018) An interactive augmented reality tool for constructing free-form modular surfaces. Autom Constr 85:135–145. https://doi.org/10.1016/j.autcon.2017.10.015
18. Gu J, Duh HBL (2011) Mobile augmented reality game engine. In: Handbook of Augmented Reality. Springer New York, pp 99–122
19. Herbert B, Ens B, Weerasinghe A, Billinghurst M, Wigley G (2018) Design considerations for combining augmented reality with intelligent tutors. Comput Graph 77:166–182. https://doi.org/10.1016/j.cag.2018.09.017
20. Juul J (2005) Half-real : video games between real rules and fictional worlds. MIT Press
21. Kitchenham B (2004) Procedures for performing systematic reviews. Keele University, Keele, UK
22. Kitchenham B, Brereton P, Budgen D, Linkman S, Almstrum VL, Pfleeger SL (1997) Evaluation and assessment in software engineering. Information & Software Technology Technol 39:731–734. https://doi.org/10.1016/S0950-5849(97)00024-4
23. Klimova A, Bilyatdinova A, Karsakov A (2018) Existing teaching practices in augmented reality. In: Procedia Computer Science. Elsevier B.V., pp 5–15
24. Klopfer E (2008) Augmented learning: research and design of mobile educational games. MIT Press, Cambridge
25. Laine TH, Nygren E, Dirin A et al (2016) Science spots AR: a platform for science learning games with augmented reality. Educ Tech Res Dev 64:507–553. https://doi.org/10.1007/s11423-015-9419-0
26. Lameras P, Arnab S, Dunwell I, Stewart C, Clarke S, Petridis P (2017) Essential features of serious games design in higher education: linking learning attributes to game mechanics. Br J Educ Technol 48:972–994. https://doi.org/10.1111/bjet.12467

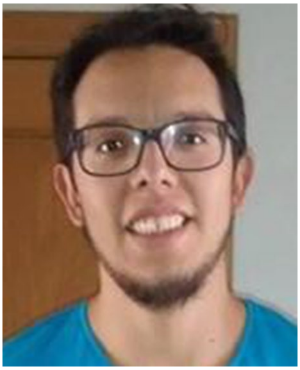
27. Liarokapis F, de Freitas S (2009) A case study of augmented reality serious games. In: Looking toward the future of technology-enhanced education: ubiquitous learning and the digital native. IGI Global, pp 178–191
28. Liarokapis F, Macan L, Malone G, Rebolledo-Mendez G, De Freitas S (2009) A pervasive augmented reality serious game. In: Proceedings of the 2009 conference in games and virtual worlds for serious applications, VS-GAMES 2009, pp 148–155
29. López-Faican L, Jaen J (2020) EmoFindAR: evaluation of a mobile multiplayer augmented reality game for primary school children. Comput Educ 149:103814. https://doi.org/10.1016/j.compedu.2020.103814
30. Maia LF, Nolêto C, Lima M, Ferreira C, Marinho C, Viana W, Trinta F (2017) LAGARTO: a LocAtion based games AuthoRing TOol enhanced with augmented reality features. Entertainment Computing 22:3–13. https://doi.org/10.1016/j.entcom.2017.05.001
31. Marin-Vega H, Alor-Hernandez G, Colombo-Mendoza LO, Sanchez-Ramirez C, Garcia-Alcaraz JL, Avelar-Sosa L (2019) Zeus – a tool for generating rule-based serious games with gamification techniques. IET Softw. https://doi.org/10.1049/iet-sen.2019.0028
32. McCallum S, Boletsis C (2013) Augmented reality & gesture-based architecture in games for the elderly. Studies in Health Technology and Informatics 189:139–144
33. Mohamed Noor N, Lob Yussof R, Yusoff FH, Ismail M (2018) Gamification and augmented reality utilization for islamic content learning: the design and development of tajweed learning. In: Communications in Computer and Information Science. Springer Verlag, pp 163–174
34. Pérez Díaz R (2016) Realidad aumentada. Tecnología para la formación. Edutec Rev Electrónica Tecnol Educ:a346–a346. https://doi.org/10.21556/edutec.2016.57.768
35. Pérez-Colado VM, Pérez-Colado IJ, Martínez-Ortiz I, Freire-Morán M, Fernández-Manjón B (2017) Simplifying location-based serious game authoring. In: Proceedings of the 5th International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM 2017) ACM international conference proceeding series. Association for Computing Machinery
36. Perry B (2015) Gamifying French language learning: a case study examining a quest-based, augmented reality Mobile learning-tool. Procedia - Soc Behav Sci 174:2308–2315. https://doi.org/10.1016/j.sbspro.2015.01.892
37. Rauschnabel PA, Rossmann A, Dieck MCT (2017) An adoption framework for mobile augmented reality games: the case of Pokémon go. Comput Hum Behav 76:276–286. https://doi.org/10.1016/j.chb.2017.07.030
38. Rosenbaum E, Klopfer E, Perry J (2007) On location learning: authentic applied science with networked augmented realities. J Sci Educ Technol 16:31–45. https://doi.org/10.1007/s10956-006-9036-0
39. Rugelj J (2016) Serious computer games Design for Active Learning in teacher education. In: Vaz de Carvalho C, Escudeiro P, Coelho A (eds) Serious games, interaction, and simulation. Springer International Publishing, Cham, pp 94–102
40. Rumiński D, Walczak K (2020) Large-scale distributed semantic augmented reality services – a performance evaluation. Graph Models 107:101027. https://doi.org/10.1016/j.gmod.2019.101027
41. Sannikov S, Zhdanov F, Chebotarev P, Rabinovich P (2015) Interactive educational content based on augmented reality and 3D visualization. In: Procedia Computer Science. Elsevier B.V., pp 720–729
42. Sauro J (2011) MeasuringU: Measuring Usability with the System Usability Scale (SUS). https://measuringu.com/sus/. Accessed 28 Jun 2020

43. Sood SK, Singh KD (2018) An optical-fog assisted EEG-based virtual reality framework for enhancing E-learning through educational games. Comput Appl Eng Educ 26:1565–1576. https://doi.org/10.1002/cae.21965
44. Sotiriou S, Bogner FX (2011) Visualizing the invisible: augmented reality as an innovative science education scheme. Adv Sci Lett 1:114–122. https://doi.org/10.1166/asl.2008.012
45. Tabuenca B, Kalz M, Ternier S, Specht M (2016) Mobile authoring of open educational resources for authentic learning scenarios. Univ Access Inf Soc 15:329–343. https://doi.org/10.1007/s10209-014-0391-y
46. Tsai C-H, Kuo Y-H, Chu K-C, Yen J-C (2015) Development and evaluation of game-based learning system using the Microsoft Kinect sensor. Int J Distrib Sensor Netw 11:498560. https://doi.org/10.1155/2015/498560
47. Vidal ECE, Ty JF, Caluya NR, Rodrigo MMT (2018) MAGIS: mobile augmented-reality games for instructional support. Interact Learn Environ Interactive Learning Environments 27(7):895–907. https://doi.org/10.1080/10494820.2018.1504305
48. Wu HK, Lee SWY, Chang HY, Liang JC (2013) Current status, opportunities and challenges of augmented reality in education. Comput Educ 62:41–49. https://doi.org/10.1016/j.compedu.2012.10.024
49. Young JC, Santoso HB (2018) Preliminary study of junoblock: marker-based augmented reality for geometry educational tool. In: Communications in Computer and Information Science. Springer Verlag, pp 219–230
50. Zsila Á, Orosz G, Bőthe B, Tóth-Király I, Király O, Griffiths M, Demetrovics Z (2018) An empirical study on the motivations underlying augmented reality games: the case of Pokémon go during and after Pokémon fever. Pers Individ Dif 133:56–66. https://doi.org/10.1016/j.paid.2017.06.024



Humberto Marín-Vega is a professor and researcher at Instituto Tecnológico Superior de Zongolica, Mexico. He received a Ph.D. degree in Engineering Sciences at Instituto Tecnológico de Orizaba, Mexico, in 2019. He received an MSc degree in Computer Systems Engineering at Instituto Tecnológico de Orizaba, Mexico. His research interests include Rich Internet Applications, Serious Games, Gamification, Augmented Reality Applications, automatic software generation, and mobile applications.

Giner Alor-Hernández is a full-time researcher of the Division of Research and Postgraduate Studies in Orizaba's Technological institute: Tecnológico de Orizaba. He received an MSc degree and a PhD degree in computer science from the Center for Research and Advanced Studies of the National Polytechnic Institute (CINVESTAV), Mexico. He is author/coauthor of around 150 journal and conference papers on computer science. His research interests include Intelligent Systems, Big Data, Internet of Things. He has been guest editor of ten JCR-indexed journals. He is an IEEE Senior Member, ACM Senior Member, Computing Mexican Academy member, and He is a National Researcher recognized by the CONACYT (SNI-2). ORCID: 0000-0003-3296-0981, Scopus Author ID: 17433252100.



Luis Omar Colombo-Mendoza is a professor and researcher at Technological Institute of Teziutlan, Mexico. He obtained a postdoctoral fellowship sponsored by the National Council of Science and Technology of Mexico (CONACYT) in 2019. He received a Ph.D. degree in Informatics from the University of Murcia, Spain, in 2017. He received an MSc degree in Computer Systems Engineering from Technological Institute of Orizaba in 2012. He is author/coauthor of around 20 journal and conference papers on Computer Science. He authored a book called Frameworks, Methodologies, and Tools for Developing Rich Internet Applications, published by IGI Global Publishing. He is a member of the National System of Researchers in Mexico (level I). His research interests include Recommender Systems, Semantic Web, Deep Learning, Cloud Computing, Internet of Things and Context-aware Systems. ORCID: 0000- 0003-3341-8422, Scopus Author ID: 55247278900.

Maritza Bustos-López received the MSC degree and a PhD degree in Engineering Sciences in Orizaba´s Technological Institute: Tecnológico de Orizaba. Her research interests include Machine Learning, Natural Language Processing, Opinion Mining, Sentiment Analysis and Recommender Systems.

Ramón Zataraín-Cabada is a Professor and Researcher of the Master of Computer Science at the Instituto Tecnológico de Culiacan. He received the Master of cience and PhD degrees in Computer Science from Florida Institute of Technology. He is a member of the National Research System in México, level II. His main research interests include e-learning, m-learning and Artificial Intelligence in the education.

Authors and Affiliations

Humberto Marín-Vega[1] · Giner Alor-Hernández[2] · Luis Omar Colombo-Mendoza[3] · Maritza Bustos-López[4] · Ramón Zataraín-Cabada[5]

Humberto Marín-Vega
humberto_marin_pd177@zongolica.tecnm.mx

Luis Omar Colombo-Mendoza
luis.cm@teziutlan.tecnm.mx

Maritza Bustos-López
maritbustos@gmail.com

Ramón Zataraín-Cabada
rzatarain@itculiacan.edu.mx

1 Tecnológico Nacional de México / I. T. S. Zongolica, Zongolica, Mexico

2 Tecnológico Nacional de México/ I. T. Orizaba, Orizaba, Mexico

3 Tecnológico Nacional de México / I. T. S. Teziutlán, Teziutlán, Mexico

4 Centro de Investigación en Inteligencia Artificial, Universidad Veracruzana, Xalapa, Mexico

5 Tecnológico Nacional de México / I. T. Culiacan, Culiacán Rosales, Mexico