Adv Mobile Learn Educ Res, 2022, 2(1): 251-262
DOI: 10.25082/AMLER.2022.01.009

RESEARCH ARTICLE

Content manager for serious games: Theoretical framework and digital platform

Anastasios Kristofer Barianos[1*] **Alexandros Papadakis**[1] **Nikolaos Vidakis**[1]

[1] Department of Electrical and Computer Engineering, Hellenic Mediterranean University, Crete, Greece

Correspondence to: Anastasios Kristofer Barianos, Department of Electrical and Computer Engineering, Hellenic Mediterranean University, Heraklion, Crete, 71410, Greece; Email: abarianos@hmu.gr

Received: January 13, 2022;
Accepted: February 14, 2022;
Published: February 18, 2022.

Citation: Barianos, A. K., Papadakis, A., & Vidakis, N. (2022). Content manager for serious games: Theoretical framework and digital platform. *Adv Mobile Learn Educ Res*, **2**(1): 251-262. https://doi.org/10.25082/AMLER.2022.01.009

Abstract: In education, digital or traditional, the material studied by students is the basis from which they will achieve knowledge. For this reason, schoolbooks have been meticulously designed to be scientifically correct and pedagogically sound. With the digitization and growth of education, and the shift to adaptive learning methods, educational content is no longer centrally decided and distributed; instead, many professionals are allowed to create and distribute material within their area of expertise, and educators often develop material for their classes. Still, educational games and their material are designed and developed similarly to books, where introducing new material by third parties is not an option. Even though techniques that could assist in the creation of adaptable, customizable gameplay and content are available and are often used in commercial games, their utilization requires technical knowledge that does not allow wide use by educators to create educational material. In this paper, we introduce a framework that will allow educators to create and manage educational content for serious games and create customized versions to cover the needs of their students and unlock the potential of adaptive learning. Towards this goal, we developed a pilot implementation of the framework that provides all the necessary tools to game developers to create adaptable games and retrieve game customizations from a server. Furthermore, a web platform has been developed where educators can browse for customizable games and custom sessions. These games and sessions have educators' custom created educational content. With our framework and supportive tools and platform, we will elevate the educational value of serious games by including educators in content creation, content manipulation, game customization and thus gameplay personalization.

Keywords: adaptive serious games, customizable games, adaptable educational content, serious games, game-based learning, Unity3D

1 Introduction

As a special category of games, Serious Games was first defined as early as 1970 by Abt (1970) as games where the explicit educational goal is the main purpose instead of entertainment. Since then, digital games have evolved from simple games with lines and dots for graphics to entire virtual 3D photorealistic worlds with endless possibilities and decisions. In modern terms, one could redefine serious games as the intersection of eLearning, game-based learning and video games. Due to confusion with "entertaining" video games, games as an educational tool have received much criticism in the past. However, it is now widely accepted that serious games carry educational value, and their potential is still unfolding (Arnab et al., 2015; Bellotti et al., 2013; Connolly et al., 2012). As a result, there is a growing interest in serious digital games and increased production and utilization. To educators, serious games motivate their pupils and carry the potential to enhance engagement and learning in a constructivist manner (Kalogiannakis & Papadakis, 2019; Papadakis et al., 2020). Production of video games is a procedure that involves many fields of expertise, bringing many individuals with different backgrounds together to create a single game and adding the educational elements amplifies this complexity, since experts of the fields, pedagogues and others must also be included in the pool of professionals involved (Breuer & Bente, 2010; Daconceicao et al., 2013). This collaboration of various areas has been proven necessary for quality improvement in all aspects that are fundamental for games (Vidakis & Charitakis, 2018). The challenge and goal of serious games are to have ludic gameplay that seamlessly coincides with learning (Huynh-kim-bang et al., 2010). Though game constituents influence the perception of any game at large, content and learning components are the foundations in this case. Creators of educational content need to consider the playful aspect of games and the appropriate structure, timing, and presentation of educational material. In other words, it is not enough to think of educational content as informative texts or dialogues within the game; instead of educational theories, methods and suitability for each student and setting must be considered, achieving an appropriate, robust environment for learning, where students will be able to assimilate the provided information and achieve knowledge (Zeeman, 2014). Though, the mixture of crucial elements does not allow for the central design of games if we also strive for global suitability and relevance. Furthermore, educators often do not trust the content provided, and thus they seek learning objects they trust and find applicable to their classrooms and students, often encouraged by school principals who wish to establish a digital learning culture that includes digital learning environments (Karakose et al., 2021). However, this introduces time consumption, insecurity and ultimately lead to the opposite result, using unfitting materials (Poultsakis et al., 2021). Under this paradigm, teachers are tasked to understand and nurture the needs of each student in their class and provide them with the learning material suitable to each while also evaluating the learning process (Kalogiannakis et al., 2021). Subsequently, most serious games lean towards improving the quality of static information and gameplay elements, overlooking the potential adaptive environments bring to education (Stewart et al., 2013; van der Vegt et al., 2016). At the same time, games are sometimes only used to add measurable reward systems and introduce some competition that will keep students interested (Kalogiannakis et al., 2021). Furthermore, video games have always been tied with the cutting edge of technology in computer science. As a result, there is a bigger interest in technical improvements of games and the introduction of new technologies, such as Virtual and Augmented reality and Artificial Intelligence, to improve learning and spark students' interest. Consequently, content, as defined above, is often neglected in a game design perspective and left to other eLearning platforms to innovate upon, especially in the case of adaptive and personalized learning (Hurtado et al., 2018). Those techniques allow students of the same class to follow different learning paths, adjust navigation, presentation or content, and thus get a unique educational experience (Katsaris & Vidakis, 2021; Brusilovsky, 2012). This approach is proven to motivate and engage students, resulting in better absorption of information and higher levels of learning (Alshammari et al., 2015). The commercial video game industry has applied similar techniques in many cases, albeit with different goals. Unfortunately, in serious games, though explorations have been attempted, they generally cover edge-cases while trying to explore the capabilities of specific techniques (Troussas et al., 2020), turning the technique into a goal instead of being the tool. Serious games are yet to become adaptable and customizable, while game-based learning is sparingly utilized in official education institutions (Papadakis et al., 2018). Therefore, we see the opportunity to apply technologies already used in other fields to serious games, generating games that can adapt to students. The goal is to facilitate educators incorporating serious games into their learning procedures, making them trust games as Learning Objects and ultimately having students engage in an experience tailored to their needs.

The first step towards this goal is to offer educators the necessary infrastructure to explore and find games they can trust or even modify their pupils' needs. A framework based on threefold principles was defined and implemented. First among the principles is asynchronous multi-discipline collaboration. Since educators can customize the game post-production, an asynchronous, anonymous collaboration is introduced. Similar to the way open-source projects might work, collaborators/contributors might not need to communicate or even be aware of all versions created, but by working on the same project and increasing its value, they do collaborate in a sense. Secondly, reusability of games and learning objects is ensured, as a many-to-many relation is now created between games and learning objects, opening up the possibility for any combination an educator might wish.

Additionally, the ratio of time spent on development and content creation for a game to the time spent playing it is increased through reusability. Lastly, adaptive content arises, although not in the true nature of the term at this stage, through the ability to offer customized versions to students. Utilizing this three-fold, educators can modify the educational content of games, making them reusable and customized to their student's needs, thus achieving all three goals. It brings power to educators similar to authoring tools while keeping designers and developers in the loop and ensuring more ludic gameplay. Thus, produced games are of higher quality and educational value, technologically relevant and more attractive to students who can enjoy the learning procedure and gain more benefits through their experience.

2 Background

Video games are among the most sophisticated products currently produced. Even though they have little to no effect on the physical sphere, compared to other industries such as medicine and civil engineering, for instance, producing video games is, in comparison, often more labour-intensive and requires collaboration from a broader spectrum. The long list of interdisciplinary professionals involved include storytellers, psychologists, marketers, programmers, graphic designers, 3D artists, game designers, etc. This list is further lengthened with pedagogues and experts in specific fields when an educational goal is to create the game. This means a wide array of workflows, tools and technologies are used by each profession and collaboration. This collaboration can make similarities emerge and create interdisciplinary innovation in many cases. To explore this interdisciplinary collaboration, any overlapping workflows and the tools used in the industry, we conducted an exploratory research of current state of the art on those matters.

2.1 Reusability & learning objects

In the software industry, the reuse of existing software to develop new solutions and products has been fundamental to the procedure since the first commercial endeavours in the field (Frakes & Kang, 2005). As a practice, using current work allows creating results faster, with a solid foundation, resulting in less time and resource spent on recurring problems and ensuring a better quality of software (Frakes & Terry, 1996). The current state of the software industry is a sparkling example of how reuse can facilitate growth. Today, any new software project will utilize several frameworks or libraries and build upon smaller, agile foundation projects. The gaming industry takes this reuse to a greater level, compared to desktop or web software, not only reusing code in the form of inheritance and frameworks, but also having entire engines for graphics, physics, or even a unified game engine, that includes the engines mentioned above and offer a large amount of groundwork for reuse.

Additionally, elements within games, such as 3D models, art and media of any kind, solutions for interactive menus and other common elements, or even entire packs including any combination of solutions, are also often reused. Their reuse has such an extent that online shops and repositories are created for sharing and distributing such elements, with great success (van der Vegt et al., 2016; van der Vegt et al., 2016). Serious games are no exception. In most cases, they are created utilizing a game engine and as many readily available assets as possible, as the labour required to create all this work is very demanding for their cost to be justified. Educational content, however, is an exception to this rule due to the specifics of each game and educational goals. This is identified as a problem within the serious games community, with the main interest in reusable educational content but not yet available. Games and Learning Alliance (GALA, 2014) has included reusability of educational components and other aspects in their roadmap for non-leisure games, effectively turning research interest towards the issue and emphasizing the value and challenge of achieving a sustainable framework for sharing educational content for games. Some similar concepts have already been circulating in the educational sphere of research, where Downes (2001) proposed Learning Objects, reusable blocks of educational material, in 2001. Downes expected the digitization of learning and recommended sharing and reusability of material commonly studied through multiple curricula. Similarly to software, he wished to achieve higher quality through the reuse and improvement of a single object, compared to having multiple objects addressing the same educational goal, as with traditional learning. The term has since faded to a large extent, but the idea behind it is primarily put to practice in daily teaching methods and protocols, such as creative commons, open-source and other sharing systems.

On the technical side of things, there have been numerous attempts, of varying success, to create structures and frameworks to utilize sharing, searching, locating and ultimately reusing educational elements for games. Most of the works propose reference systems, with little to no concern over the technical aspects and infrastructure required for reusability. Outstanding works in the field are the RAGE project (Georgiev et al., 2016) and the Serious Games Society (Serious Game Society, 2019). RAGE aims to create absorptive, extendable, portable and, above all, reusable assets with minimum dependencies, explicitly designed for serious games. The goal of RAGE is to decrease the resources invested by small studios to produce serious games while improving the quality of the final product (van der Vegt et al., 2016; van der Vegt et al., 2016). The Serious Games Society has also produced various tools and systems to facilitate serious games production. Those tools include a, now abandoned, catalogue of web services, a reference system to identify assets shared in the society, and a framework for learning analytics, called GLEANER (Dagnino et al., 2015). GLEANER has been included in the RAGE project, and with further developments became RAGE Analytics (Serious Game Society, 2018). The reference system developed, namely Serious Games Reusability Point of Reference (Serious Game Society, 2019), is an effort towards reusability of serious games assets, where educational content is not the primary concern but could be shared if modelled as an asset. Despite its merits, drawbacks include the lack of guarantees of quality or portability and no infrastructure provided to assist developers in adopting available assets. Alternatively, any established specification for asset creation and sharing is necessary. Those drawbacks are arguably why the reference never gained much traction within the community. Thus, we can argue that RAGE is a more complete and unified proposal in the field of reusability within educational games. Nevertheless, RAGE focuses on game development problems but excludes the most fundamental thing about educational games, knowledge. Educational content is overlooked by the RAGE framework, except if modelled into specific assets utilized by the framework.

2.2 Dynamic content

Personalized learning has gained popularity in research lately due to the consensus that each student is unique in the way she learns (Akkoyunlu & Soylu, 2008). Since the dawn of digitized learning, techniques such as Adaptive Hypermedia have been identified as a powerful tool that provides personalized learning paths. According to this technique, each student's content is selected from a pool of available material after applying specific filters dependent on personal preferences, background, and educational profile (Mulwa et al., 2010). With this model for learning, diversity between students is respected, and inclusive learning is included in the learning process [37]. Though these techniques are growing in popularity among many online learning systems, educational games have yet to follow this scheme, except for some experimental work. Most notably, ALIGN (Peirce et al., 2008) was maybe the most successful attempt to introduce this workflow into serious games. However, to achieve this, ALIGN was reliant on a complex architecture that discouraged game developers from utilizing ALIGN and creating adaptive games. As a result, the creators concluded that accomplishing adaptive educational games through the Adaptive Hypermedia logic would require substantial financial and technical resources, thus impeding advances in the field (Peirce et al., 2008). The idea of utilizing techniques from Adaptive Hypermedia was, as a result, abandoned in general, and alternative techniques were instead introduced. The IOLAOS platform (Vidakis & Charitakis, 2018) offers an interesting point of view on the technical side of adaptivity.

IOLAOS offers several web services to make usage portable with minimal difficulties. Game developers can access valuable information about the current players through those web services, such as age, learning style, learning difficulties, and more. With this information in mind, game designers can model their game to have any adaptability needed for the circumstances, create all the scenarios and know-how that each variable will affect game elements and narrative. Despite the large facilitation towards adaptive games IOLAOS offers with the available web services, there are specifications, or tools, that call for specific forms or workflows to create adaptive games. Thus, educational content creators and game developers have +to develop the systems to collaborate and make the content available to each game. In practice, this means that time needs to be spent designing and developing collaboration software and management systems. Each team creates their workflow and management system, but it is only applicable in the short term for specific projects, with no availability to third parties, such as educators using those games with their students.

Furthermore, such collaborations might produce adaptive games based on "dynamic" content chains (Karagiannidis et al., 2001). However, those chains only utilize a finite, predetermined pool of available content. What is needed to achieve authentic, dynamic content is to enrich the content pool openly, without the active participation of a development team and the release of new versions.

3 Our proposal

As a result of our field investigation, we discovered that serious game content is underrepresented in research and development activities. The current technological reality has a lot to offer regarding high-quality material, enhanced development, management, collaboration techniques, and increased freedom for all professions involved in the procedure. We propose a framework to support dynamic educational content management and use it within serious games to explore the research topic and as a foundation for future attempts. This framework will provide truly dynamic content to serious games and improve material by taking a different production approach independent of the game's initial design and development. In further detail, we begin by embracing Vidakis & Charitakis (2018) assumption that each professional possesses highly specialized skills and talents that cannot be disregarded or replaced. For instance, if we look at Authoring Tools, we examine an excellent set of tools for educators to create interactive material. However, the produced material is very limited in experiences provided, compared to games with the level of attraction that gaming studios can create.

On the other hand, gaming studios do not have the capacity, staff, or experience to create and include educational material of high quality, personalized to each class and student. Even when collaborating with educators, educational authors and experts of the field, the result is lacking in comparison to the material each educator creates for her class, with knowledge of the strengths and difficulties of each student in the class. However, like all collaborations, many resources and time are required. In this day and age of computer-assisted collaborations, any interdisciplinary field should effectively use technology to create better collaboration conditions that require less from all parties while delivering superior results. Furthermore, content must be enriched and modified in a manner that does not involve the developer and is immediately available, with no need for software updates and new releases. Thus, we envision a set of specifications and frameworks for creating and managing educational content, catering to the needs of all stakeholders. Metadata will always be included, complementing the content and allowing for appropriate use through filtering and selection processes. To that end, a framework will specify rules and provide assets for integration into games, allowing content to be dynamically coupled to the game at run time. This pairing will be based on game and player as parameters, selecting only content appropriate for the active storyline and theme while also meeting the demands of the current player and the educational goals of the session. After the selection process, the content will be added to the game in a modular fashion and presented in place of the default content. Web services and a content repository will facilitate this process.

Furthermore, infrastructure for educators to submit educational materials is required and provided on a website. A content editor will be available on this website, from which educators can create, select, and limit details and content for their learning sessions with their students. The online services make gaming sessions dependent on internet connectivity, which is a problem that must be anticipated and addressed throughout development. A possible solution to this problem could be some “educational packets” that could be shared in various on and offline methods, saved to a local folder and used when internet connectivity is limited or unavailable. Figure 1 illustrates a simplified overview of the system architecture.

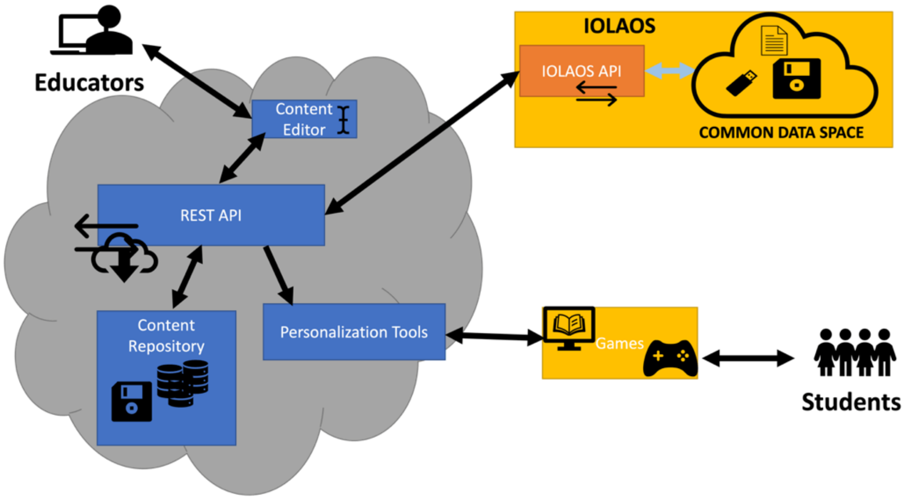

Figure 1 System overview

The entire system is comprised of the content editor, the RESTfull API, a content repository a set of personalization tools. Additionally, IOLAOS and the Games are included as external elements that are crucial in realizing or triggering procedures. IOLAOS provides user profiling through its web services that provide user preferences and Learning Styles, which can be a functional in-game adaptation. The educators use the Content Editor to author instructional materials and customize the content in future class sessions. The content and specific adjustments selected will be saved to the content repository, which has a non-relational database and a file storage server, through appropriate web services provided by the RESTful API. This same content is later retrieved through the RESTful API for use by the personalization tools that will change the content of games at runtime.

Additionally, the RESTful API acts as a mediator between the IOLAOS and the system for authorization purposes and retrieves player preferences and other data needed by the personalization tools. Thus, when students access a game, they provide their credentials, authenticated following a flow from the game to the personalization tools, triggering a request to IOLAOS API through our REST API. After this authentication, the personalization tools request player preferences and profiling available on IOLAOS for each student through the REST API. Having player data available, the personalization tools then make another request to retrieve appropriate content from the repository and modify the default content adding or replacing it with the custom content appropriate for the learning session and current student.

4 Pilot implementation

We constructed a pilot implementation as part of this endeavour, including the most crucial technological parts of the architecture proposed above. We concentrated on approaches for making games adaptive at runtime and the inextricable parts, such as the necessary parts of the content repository, critical web services for this portion of the implementation, and an initial version of the content editor. Developing a game adaptation engine is a difficult task, and thus we considered the personalization tool the most challenging part of our architecture and the priority. The challenge stems from the fact that each game is unique as it can be developed using a variety of commercial or custom game engines or even with more straightforward approaches, lacking a game engine. Additionally, a broad spectrum of formats is used, as there are significant differences in requirements, assets, and implementations for different target platforms (such as desktop, web and mobile), operating systems, etc. We assumed all games would rely on a basic architecture that will include a core, similar to game engines, for our initial endeavour. This core could differ widely between implementations, but we consider it an abstract commonality to build our specification. Our specifications scope is above this core, where we create a content adaptation module and the protocol of communication with the REST API. This architecture is presented in Figure 2, Generalized Architecture. With this architecture, we expect the Game Engine Core to be designed to allow the Content Adaptation Module to identify the adaptable components of the game. Assuming this requirement is fulfilled, the Content Adaptation Module uses standard HTTP requests through the Requests Handler to communicate with the RESTful API to retrieve the material for the current player and session. When the material is available to the Content Adaptation Module, it changes the default content loaded by the Game Engine Core with the new material retrieved through this procedure. The Requests Handler is part of our architecture, even though most engines and platforms have this capability by default, as we would like to make this a modular extension, and thus we need a communication layer specific to the Content Adaptation Module.

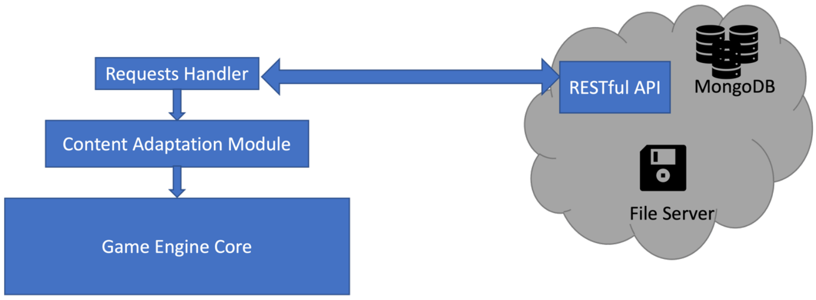

Figure 2 Generalized architecture

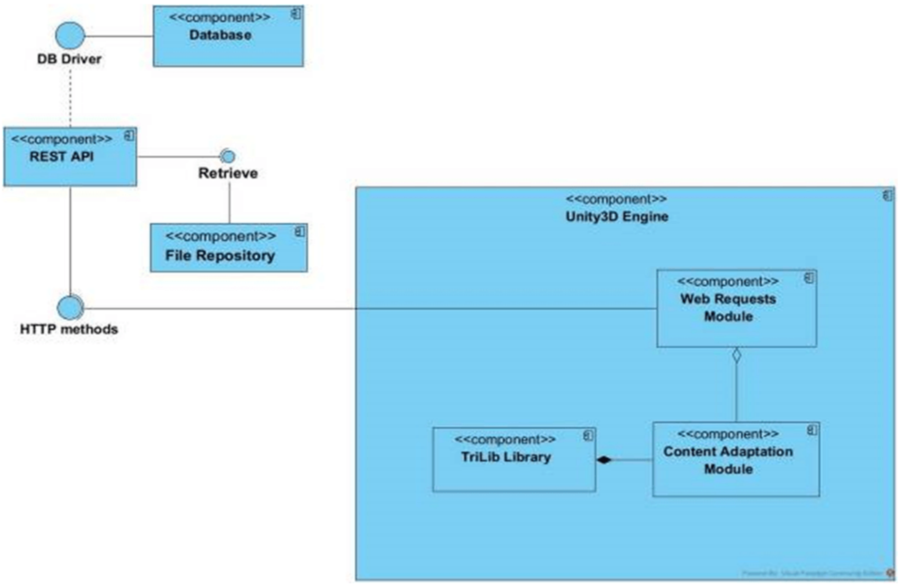

Figure 3 System architecture for unity implementation

We chose to realize the aforementioned generalized architecture for a popular game engine for our pilot implementation. We selected Unity3D Realtime Platform (Technologies, 2020) due to its popularity among the serious game community and students and the authors' prior experience with the engine. In addition, we chose a serious game, namely ThimelEdu (Vidakis et al., 2019), to be the subject of adaptation since modifying an existing game would be a more appropriate and realistic situation than creating a mini-game with the sole purpose of realizing our architecture. With this specific technical background, we further defined our generalized architecture into a specific one for this particular case, illustrated in Figure 3, System Architecture for Unity Implementation. Our Content Adaptation Module has to become part of the Unity engine to ensure efficiency and modify assets at runtime. It is thus implemented in C# and used as an asset itself, a module for the game. The Web Requests Module is also implemented within the same asset and thus becomes part of the engine. The last significant addition is the TriLib Library (Reis), which simplifies importing and rendering 3D models at runtime. We utilized NodeJS to create the API, ReactJS to create the web page hosting the Content Editor and MongoDB for our database for the other parts of the architecture.

4.1 Components

4.1.1 Web request module

As mentioned above, we wanted to create this module to overcome any unpredictability between engines, have as few dependencies as possible and keep a separate layer of communication that only serves our adaptation system. Thus, we created a class that implements Web Requests, with predefined specifics, such as headers and JSON format, for our and IOLAOS APIs, thus simplifying the procedure for developers that will use the system in the future. Additionally, our module follows the asynchronous paradigm, usually utilized in HTTP communication. With Unity, coroutines are suggested when working with Web Requests. Coroutines are closely tied to the way the Unity engine works. They span multiple frames while returning program flow control to the location they got called from (Unity3D, 2021). This is very useful when calculating game logic but is less than ideal when working with remote resources.

Drawbacks include the creation of massive monolithic logic; the synchronous flow of events and error handling becomes a complication since we need to inspect asynchronous data synchronously. Those flaws might cause a game to freeze while waiting for a response. Usually, this is not noticed by the user, as such exchanges happen in a matter of milliseconds; however, network communications can be unpredictable and should not be relied upon to this level. On the other hand, the async paradigm pauses the execution of a function and releases the thread, allowing the program to continue execution while waiting for the response. An incomplete task is returned to the function's caller, who continues the program's execution. When the scheduled event occurs, the async method will continue to run (in this case, the response from the server has been received). Thus, we decided to create our module following the asynchronous approach. It should also be noted that the functionality is not tied to the two APIs the module is designed for, it can be used with any other web services that utilize the same parameters, which are very common, and it can be parameterized to work with others as well. However, we considered this out of scope, and there was no reason to restrict usage.

4.1.2 Content adaptation module

In Unity, four mechanisms can be used to import assets at runtime. Namely, the mechanisms are Asset Bundles, Addressable Asset System, Streaming Assets and Resources. Each has advantages and disadvantages, and which strategy to use is determined by the specific conditions of each project and the problem to be handled. In our implementation, we utilized the Streaming Assets as a folder where we download our resources to then import into the engine. However, one challenge is faced when handling 3D models, in which case all of the available methods require the model to be in a Unity proprietary format, which essentially means that a developer has to prepare the model as an asset before it can be used in our scope. We want to decouple the process from the game developer, so 3D models found on the internet or created by anyone interested should be able to be used without the additional workflows and technicalities that come with the need to create a unity asset before educators, and other stakeholders can use it.

For this reason, we utilized the TriLib library as a tool that allows us to dynamically load 3D models of various formats at runtime in the Unity3D engine. The flow and communications developed to make runtime changes are presented in Figure 4, Scene Personalization Flow. Unity splits game logic into blocks called scenes, where the entire environment and everything within it can be changed. On this ground, the scene is the basic block for our system, and we have a Scene Personalizer that will communicate with the Engine Core to identify all the objects that can potentially be changed. The scene personalizer utilizes the Request Handler with this information, sending a request for dynamic content based on the current player and game. The response is in JSON form and can include multiple assets of various formats. The Scene Personalizer parses the response and handles each asset individually, creating C# objects and downloading any necessary files to the Streaming Assets folder. Texts, being the most straightforward kind of content, is replaced in the corresponding variables or data structures, and any other files are imported from the streaming assets and then replaced with their corresponding in-game asset. 3D models, a more significant challenge, are processed by a model importer, which presets variables for the TriLIb Handler and handles the replacement of in-game models when the import is complete by TriLIb.

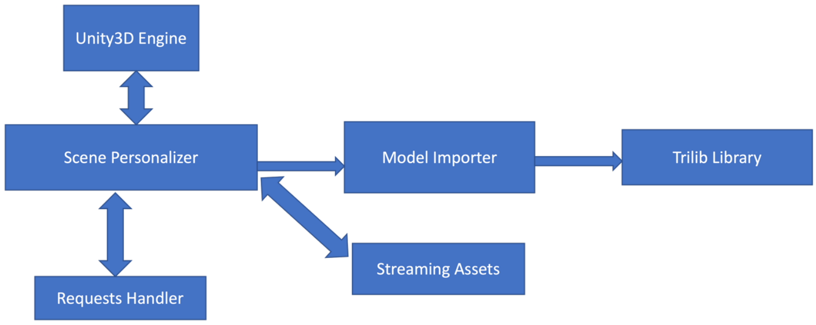

Figure 4 Scene personalization flow

5 Results

As a brief showcase of the results of our framework, we can examine a concise scenario in which a specific Learning Object is modified. In this case, the default in-game Learning Object teaches the cothurnus, used in ancient Greek theatre. As seen in Figure 5, Default Game Contents, the default setup includes a 3D object of a cothurnus, which when interacted with opens a dialogue presenting three drawings of cothurnus and a brief text stating, "*The Cothurnus were the shoes used by tragic actors. They. Were tall, had soles that made the actors look taller and were usually built from wood*". Using our content editor, an educator changes the learning object by replacing the 3D object, the images and the text to instead teach about sandals. The content editor with the additions the educator does can be seen in Figure 7, Content Editor. The in-game results are showcased in Figure 6, Customized Game Contents, where we can see the difference compared to the default content. A 3D sandal replaces the 3D cothurnus, the three drawings are replaced by images of sandals and a statue, and the text now reads, "*Sandals of the ancient time were distinguished into two kinds, baxea and cothurnus. The first was simpler sandals, made from willow leaves and twigs, usually worn in comedy plays. The latter was more like a boot, with thick soles to add to the stature and was usually worn in tragedy plays*". All material presented is provided by the educator in the content editor. The technological opportunities are discernible even with the limited scope and the game's limitations. Future serious games designed with adaptivity in mind will provide new levels of content customization with dynamic material deeply integrated with the game design.

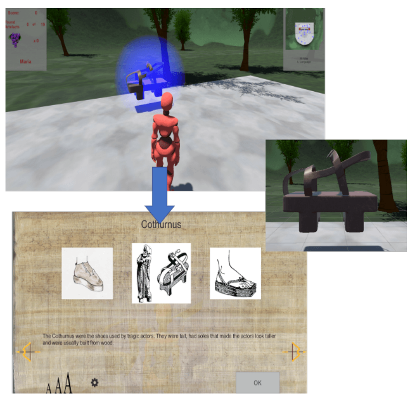

Figure 5 Default game contents

Figure 6 Customized game contents

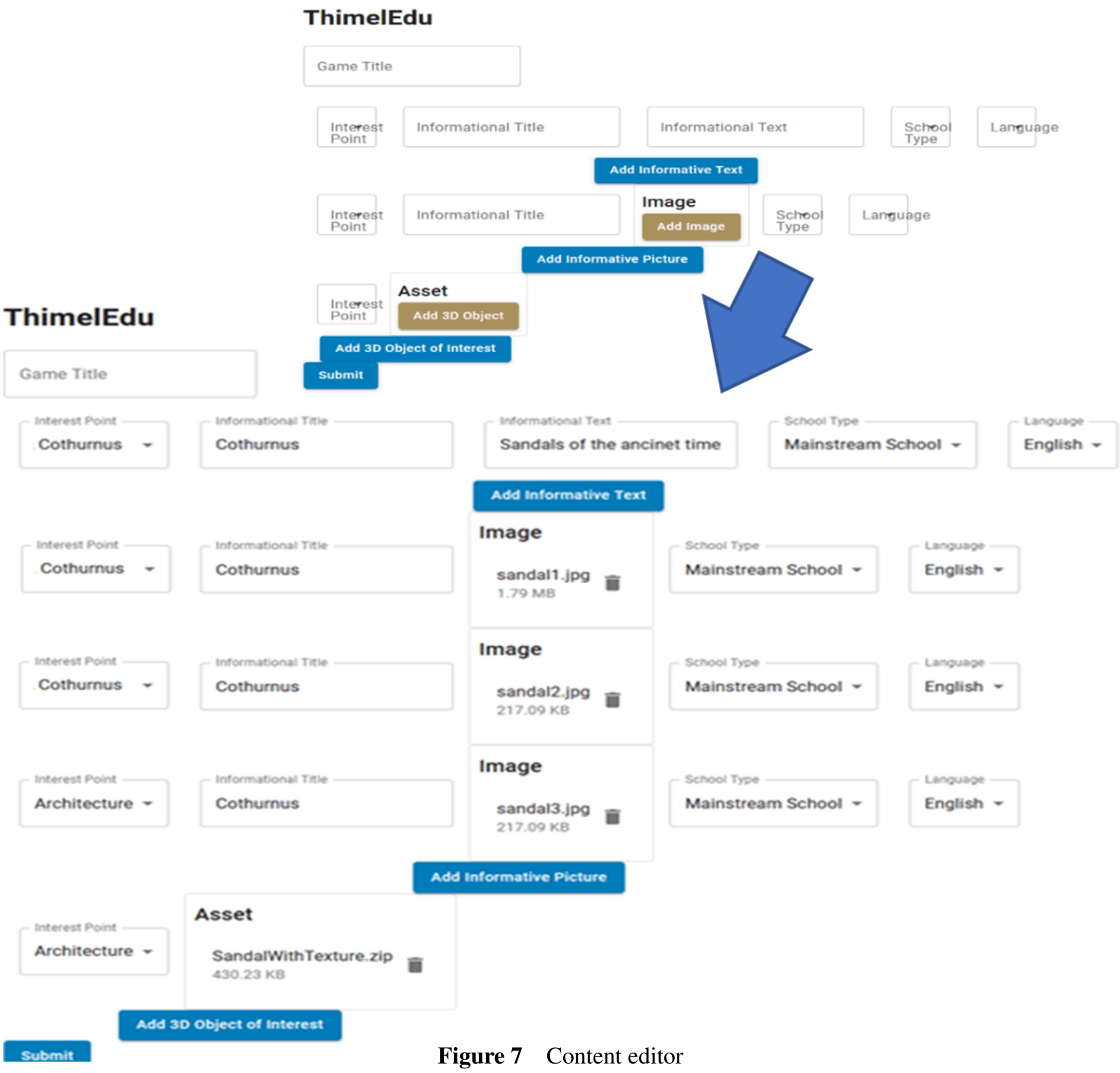

Figure 7 Content editor

6 Conclusion and future work

Education is rapidly changing and catching up with technological advancements. Serious games are patterning regularly, and game-based learning is becoming increasingly prevalent in formal and informal learning contexts. On the other hand, serious games have yet to be conceptually characterized. Large-scale research and discussions are held to speculate and test ideas to propel the discipline ahead. The research usually focuses on lofty aims that are just out of reach, often bringing them slightly closer to reality. However, we frequently overlook considerably more realistic goals in this process and have a more immediate impact. Game production, particularly educational development, necessitates a wide range of expertise and disciplines.

Additionally, creating a game that is adaptable enough to be helpful for any learner, regardless of background or circumstance, is a difficult task. Through this endeavour, we identified a shortcoming in the serious game industry, the lack of current technology utilization with the goal of runtime adaptation of games. We used this insight to develop an abstract idea on which we built the requirements and design of an innovative project that addresses the issue.

Furthermore, we created a pilot implementation, through which we realized the concept and have a foundation for future extensions that could bring the entirety of the framework to life. We expect the suggested framework to improve the learning process and benefit students while supporting educators and developers in their collaboration. Additionally, we hope the framework serves as a guideline and specification for content authoring and game development, effectively reducing resources required for serious games publishing while improving game quality, educational value, and player engagement. Limitations of the study that should be pointed out are the lack of experimental data in the field and the narrow implementation of the framework that does not reflect the entirety of the concept.

Unfortunately, our pilot implementation is limited to function with only one specific game engine and thus it is one of our priorities in future works to build the infrastructure for additional game engines. Additionally, we wish to improve the framework to be applicable to web-based games. Furthermore, our primary focus in the future will be on expanding the Content Editor and REST API, improving the experience for educators and offering additional capabilities. Likewise, we want to improve the current implementation by converting the methods that load the remote object to Async procedures, switching the entire solution to an asynchronous paradigm, and optimizing the design. Finally, we want to use semantic notation with the data handled by the framework so that future advancements can be applied in automation and tracking.

References

Abt, C. C. (1970). Serious Games.
https://doi.org/10.1109/VS-GAMES.2009.8

Akkoyunlu, B., & Soylu, M. Y. (2008). A study of student's perceptions in a blended learning environment based on different learning styles. Educational Technology and Society, 11(1), 183-193.
https://doi.org/10.1007/s00217-010-1351-2

Alshammari, M., Anane, R., & Hendle, R. J. (2015). An E-learning investigation into learning style adaptivity. Proceedings of the Annual Hawaii International Conference on System Sciences, 2015-March, 11-20.
https://doi.org/10.1109/HICSS.2015.13

Arnab, S., Lim, T., Carvalho, M. B., Bellotti, F., de Freitas, S., Louchart, S., Suttie, N., Berta, R., & de Gloria, A. (2015). Mapping learning and game mechanics for serious games analysis. British Journal of Educational Technology, 46(2), 391-411.
https://doi.org/10.1111/bjet.12113

Bellotti, F., Kapralos, B., Lee, K., Moreno-Ger, P., & Berta, R. (2013). Assessment in and of serious games: An overview. Advances in Human-Computer Interaction, 2013.
https://doi.org/10.1155/2013/136864

Breuer, J., & Bente, G. (2010). Why so serious? On the relation of serious games and learning. Eludamos. Journal for Computer Game Culture, 4(1), 7-24.

Brusilovsky, P. (2012). Adaptive hypermedia for education and training. Adaptive Technologies for Training and Education, 46-66.
https://doi.org/10.1017/CBO9781139049580.006

Connolly, T. M., Boyle, E. A., Macarthur, E., Hainey, T., & Boyle, J. M. (2012). A systematic literature review of empirical evidence on computer games and serious games. Computers & Education, 59(2), 661-686.
https://doi.org/10.1016/j.compedu.2012.03.004

Daconceicao, R., Locke, C., Cooper, K., & Longstreet, C. S. (2013). Semi-automated serious educational game generation: A component-based game engineering approach. Proceedings of CGAMES 2013 USA - 18th International Conference on Computer Games: AI, Animation, Mobile, Interactive Multimedia, Educational and Serious Games, 222-227.
https://doi.org/10.1109/CGames.2013.6632637

Dagnino, F. M., Ott, M., Pozzi, F., & Yilmaz, E. (2015). Serious Games Design: Reflections from an Experience in the Field of Intangible Heritage Education. The 11th International Scientific Conference ELearning and Software for Education, 2015, 57-64.
https://doi.org/10.12753/2066-026X-13-131

Downes, S. (2001). Learning Objects: Resources For Distance Education Worldwide The Need for and Nature of Learning Objects. International Review of Research in Open and Distance Learning, 2(1), 1-35.
https://doi.org/10.19173/irrodl.v2i1.32

Frakes, W. B., & Kang, K. (2005). Software reuse research: Status and future. IEEE Transactions on Software Engineering, 31(7), 529-536.
https://doi.org/10.1109/TSE.2005.85

Frakes, W., & Terry, C. (1996). Software reuse: metrics and models. ACM Computing Surveys, 28(2), 415-435.
https://doi.org/10.1145/234528.234531

GALA. (2014). Roadmap on Serious Games. 1-33.
https://doi.org/10.1155/2014/358152

Georgiev, A., Grigorov, A., Bontchev, B., Boytchev, P., Stefanov, K., Bahreini, K., Nyamsuren, E., van der Vegt, W., Westera, W., Prada, R., Hollins, P., & Moreno, P. (2016). The RAGE Software Asset Model and Metadata Model, 191-203.
https://doi.org/10.1007/978-3-319-45841-0_18

Hurtado, C., Licea, G., & Garcia-Valdez, M. (2018). Integrating Learning Styles in an Adaptive Hypermedia System with Adaptive Resources. Studies in Systems, Decision and Control, 143, 49-67.
https://doi.org/10.1007/978-3-319-74060-7_3

Huynh-kim-bang, B., Wisdom, J., & Labat, J. (2010). Design Patterns in Serious Games: A Blue Print for Combining Fun and Learning Introduction: Making Learning Fun. Journal for Computer Game Culture, 1-18.
https://doi.org/10.1080/0142569880090306

Kalogiannakis, M., & Papadakis, S. (2019). Evaluating pre-service kindergarten teachers' intention to adopt and use tablets into teaching practice for natural sciences. International Journal of Mobile Learning and Organisation, 13(1), 113.
https://doi.org/10.1504/IJMLO.2019.096479

Kalogiannakis, M., Papadakis, S., & Zourmpakis, A.-I. (2021). Gamification in Science Education. A Systematic Review of the Literature. Education Sciences, 11(1), 22.
https://doi.org/10.3390/educsci11010022

Karagiannidis, C., Sampson, D., & Cardinali, F. (2001). Integrating adaptive educational content into different courses and curricula. Educational Technology and Society, 4(3), 37-44.

Karakose, T., Polat, H., & Papadakis, S. (2021). Examining Teachers' Perspectives on School Principals' Digital Leadership Roles and Technology Capabilities during the COVID-19 Pandemic. Sustainability, 13(23), 13448.
https://doi.org/10.3390/su132313448

Katsaris, I., & Vidakis, N. (2021). Adaptive e-learning systems through learning styles: A review of the literature. Advances in Mobile Learning Educational Research, 1(2), 124-145.
https://doi.org/10.25082/AMLER.2021.02.007

Mulwa, C., Lawless, S., Sharp, M., Arnedillo-Sanchez, I., & Wade, V. (2010). Adaptive educational hypermedia systems in technology enhanced learning. Proceedings of the 2010 ACM Conference on Information Technology Education - SIGITE '10, 73.
https://doi.org/10.1145/1867651.1867672

Papadakis, S., Kalogiannakis, M., & Zaranis, N. (2018). Educational apps from the Android Google Play for Greek preschoolers: A systematic review. Computers & Education, 116, 139-160.
https://doi.org/10.1016/j.compedu.2017.09.007

Papadakis, S., Trampas, A., Barianos, A., Kalogiannakis, M., & Vidakis, N. (2020). Evaluating the Learning Process: The "ThimelEdu" Educational Game Case Study. Proceedings of the 12th International Conference on Computer Supported Education, 290-298.
https://doi.org/10.5220/0009379902900298

Peirce, N., Conlan, O., & Wade, V. (2008). Adaptive educational games: Providing non-invasive personalized learning experiences. Proceedings - 2nd IEEE International Conference on Digital Game and Intelligent Toy Enhanced Learning, DIGITEL 2008, 28-35.
https://doi.org/10.1109/DIGITEL.2008.30

Poultsakis, S., Papadakis, S., Kalogiannakis, M., & Psycharis, S. (2021). The management of Digital Learning Objects of Natural Sciences and Digital Experiment Simulation Tools by teachers. Advances in Mobile Learning Educational Research, 1(2), 58-71.
https://doi.org/10.25082/AMLER.2021.02.002

Reis, R. (n.d.). TriLib 2.
https://ricardoreis.net/trilib-2

Serious Game Society. (n.d.-a). GLEANER. Retrieved January 27, 2018.
https://e-ucm.github.io/gleaner

Serious Game Society. (n.d.-b). Serious Game Society. Retrieved January 25, 2019,
https://seriousgamessociety.org/about

Serious Game Society. (n.d.-c). SGREF. Retrieved January 27, 2019,
http://www.sgref.com

Stewart, J., Lizzy, I., Looy, J. van, Mariën, I., All, A., Schurmans, D., Willaert, K., Grove, F. de, Jacobs, A., & Misuraca, G. (2013). The Potential of Digital Games for Empowerment and Social Inclusion of Groups at Risk of Social and Economic Exclusion: Evidence and Opportunity for Policy.
https://doi.org/10.2791/88148

Technologies, U. (n.d.). Unity 3D Real-Time Development Platform. Retrieved May 16, 2020,
https://unity.com

Troussas, C., Krouska, A., & Sgouropoulou, C. (2020). Collaboration and fuzzy-modeled personalization for mobile game-based learning in higher education. Computers & Education, 144, 103698.
https://doi.org/10.1016/j.compedu.2019.103698

Unity3D. (n.d.). Unity Scripting Reference. Retrieved May 18, 2021,
https://docs.unity3d.com/ScriptReference/index.html

van der Vegt, W., Nyamsuren, E., & Westera, W. (2016). RAGE Reusable Game Software Components and Their Integration into Serious Game Engines (pp. 165-180).
https://doi.org/10.1007/978-3-319-35122-3_12

van der Vegt, W., Westera, W., Nyamsuren, E., Georgiev, A., & Ortiz, I. M. (2016). RAGE Architecture for Reusable Serious Gaming Technology Components. International Journal of Computer Games Technology, 2016, 1-10.
https://doi.org/10.1155/2016/5680526

Vidakis, N., Barianos, A., Trampas, A., Papadakis, S., Kalogiannakis, M., & Vassilakis, K. (2019). Generating Education in-Game Data: The Case of an Ancient Theatre Serious Game. Proceedings of the 11th International Conference on Computer Supported Education, 36-43.
https://doi.org/10.5220/0007810800360043

Vidakis, N., & Charitakis, S. (2018). Designing the Learning Process. Proceedings of the 10th International Conference on Subject-Oriented Business Process Management - S-BPM One '18, 1-11. https://doi.org/10.1145/3178248.3178254

Zeeman, M. (2014). Factors that Influence Computer Science Students' Attitude towards Serious Games in the Classroom. Mediterranean Journal of Social Sciences. https://doi.org/10.5901/mjss.2014.v5n21p413