UEmbed: An Authoring Tool to Make Game Development Accessible for Users Without Knowledge of Coding

Imran Hossain, Lino Weist, Sarah Voß-Nakkour
studiumdigitale, Goethe University Frankfurt
Frankfurt am Main, Germany
e-mail: (hossain / weist / voss)@studiumdigitale.uni-frankfurt.de

***Abstract*—There has been a rising interest for Serious Games in recent years, which lead to more and more Game Engines becoming available for everyone to use. Although this availability is very welcoming, the usage on the other hand is sadly not that accessible. There is a dearth of knowledge on the technical side needed to have an usable output. This problem excludes those who are brimming with ideas to move the needle in their departments with Serious Games but lack the insight on the useability of these Game Engines. This makes it a daunting task to design and develop a Serious Games as this problem can only be solved with time, knowledge, new hires and a moderate budget. Thus, an Open Source Authoring Tool named UEmbed is being developed. It would make the task of creating and developing a game with the latest technology of the Unreal Engine very accessible for everyone, including individuals with cognitive impairment or dyscalculia. Without any prior knowledge, it would be possible to design and set the environments, dialogues, characters for an immersive Serious Game with the help of a Browser-based Authoring Tool in conjunction with a pre prepped project. Also, a template scene with characters and story will be provided to showcase the individual functionalities. This tool can be used to showcase various scenarios with a high-end graphical fidelity. With this open source project concept, it will be accessible to anyone and it can also be modified and further extended by the community. This approach will ensure that any institute or individuals with a tight budget will have access to develop a product on par with the latest technological trends.**

Keywords- UEmbed; Serious Game; Authoring Tool; Accessibility.

I. INTRODUCTION

Digital accessibility has become a critical issue in recent years, with millions of people around the world facing barriers to accessing digital products and services. In 2019, the European Union took a bold step to address this issue by adopting the European Accessibility Act (EAA). This directive aims to improve the functioning of the internal market for accessible products and services by removing barriers created by divergent rules across member states. The EAA has brought digital accessibility to the forefront of national agendas across Europe, signaling a major shift in how we approach inclusivity in the digital age. Germany has made significant progress towards digital accessibility by implementing the Barrierefreiheitsstärkungsgesetz (BFSG), which is in line with the European Accessibility Act (EAA). This groundbreaking legislation demonstrates Germany's unwavering commitment to promoting inclusivity and ensuring that digital products and services are accessible to everyone, regardless of ability or disability. The EAA enshrines accessibility as a fundamental right and empowers individuals to fully participate in the digital economy.

Discussions regarding this topic have been held at universities for quite some time. Therefore, the idea occurred to approach this topic in the form of Serious Games.

Section 2 defines the problems of serious game development. Since comparable tools are often more complex to use and offer less impressive graphics, a comparison of the state of the art is made in Section 3. The design and technical aspects of the authoring tool are described in detail in Section 4. Section 5 presents the workflow from the user's perspective. The paper concludes with a short summary and an outlook on the future of this project in Section 6.

II. SERIOUS GAMES

According to R. Dörner, S. Göbel, W. Effelsberg, and J. Wiemeyer (2016), a serious game is a digital game created with the intention to entertain and to achieve at least one additional goal [1]. These games can have an additional goal to teach players about certain topics in a playful way. The game "Lola's First Semester" [2], developed by studiumdigitale, the central e-learning unit of the Goethe University Frankfurt, uses the university as a scenario and illustrates the everyday life of a visually impaired student named Lola. Players are put in a situation where they have to cope with various everyday challenges from Lola's point of view and learn to empathize with the situation of a visually impaired person. The goal was to make the players more aware of the different problems such a person might face. This was done through a series of mini-games, each dealing with a specific situation in the life of the protagonist "Lola" at university. For example, these mini-games highlighted the inaccessible design of PowerPoint slides or PDFs, or the criteria to consider for accessible videoconferencing. At the end of the game, the players were asked about their experiences during the game and the results were evaluated in a study [3].

During the design, programming, and publishing phases of this serious game about accessibility, several issues arose throughout the process. It became clear that creating a serious game from the ground up requires a significant amount of time, budget, and manpower. The question arose as to how best to allocate the available expertise. The game industry and the usual processes for creating a game quickly came to mind. Typically, game engines are used for programming. However, as programming is too complex for many university staff, the previously identified problem areas (time, budget and staff) would hardly be alleviated. There are game engines that can be used for free under certain conditions, but they

 ISBN: 978-1-68558-084-1

still require a lot of time and manpower without specialized knowledge. This led to the idea of an authoring tool. This tool should allow people without programming knowledge to use the possibilities of complex game engines. Ideally, everyone would have the chance to create their own game, whether it's a serious game or just for entertainment. This tool should also be accessible and usable outside the university.

Part of an authoring tool was programmed where users can click on buttons on a web interface to summon buildings, trees, and characters on a top-down map and also change some basic properties such as name or size. This construction can then be exported as a JSON file that can be imported into the Unreal Engine and rebuilt in 3D.

III. State of the Art

The number of video game players is constantly growing [4], and there has also been significant growth in the serious games sector, particularly in academia [5]. This increased interest in the medium has led to some changes in the gaming industry. It has become much easier for gamers and other interested parties to develop their own games on their home computers. There are many reasons for this, one of which is easier access to game engines. The history of the availability of the Unreal Engine can be used as an example to illustrate this. In November 2009, the Unreal Engine 3 was made available for free for non-commercial use. You could download and use the Unreal Development Kit for free, but you could not sell the result. In early 2015, the Unreal Engine could also be used for free for commercial purposes, but only up to a certain revenue limit (3,000 USD / quarter). Another five years later, in May 2020, the licensing model was changed again, allowing the engine to be used completely free of charge if the gross revenue does not exceed one million US dollars. This development shows how easy it has become for small projects and studios to implement the use of professional game engines. Other large game engines, such as Unity, have also become more accessible over time. The use of such engines is no longer limited to large companies. Now anyone can download and use them as they see fit.

However, using game engines requires good programming skills. As a result, many people are prevented from implementing their own game ideas simply because they do not have the time to acquire or train the necessary skills. Other developers have recognized this problem and implemented tools to give people without programming skills access to professional game engines. In the following, we will discuss some of these tools and explain how they differ from the UEmbed vision.

A. RPG Maker

This tool has been available since the 1990s and has been continuously developed. The games created with this tool have pixelated graphics from a top-down perspective, like the games from the mid-80s to early 90s. However, it is somewhat limited in design as it is mainly designed to create role-playing games (RPGs). Although no programming skills are required, the interface has many menu items and submenus, making the tool very complex. As a result, it is more suitable for users who are familiar with computers and also requires a longer learning curve. However, to fully exploit its potential, programming skills are still required. Although not mandatory, they provide more options in the design of the game. Therefore, it is not suitable as a tool for people without programming skills. It is also not free, although the price is quite low. In addition, it limits the implementation of game ideas. Because of the perspective and the limited interaction possibilities, only certain types of games can be implemented. It would be desirable to have more freedom in implementation and less restrictions on creativity.

B. CoSpaces Edu

CoSpaces Edu is a tool that comes very close to the idea of UEmbed. With CoSpaces Edu, users can create a game with 3D graphics. These games are heavily focused on Virtual Reality (VR), although they can also be used without VR equipment. However, playing the games outside of VR requires knowledge of certain gaming conventions, such as controlling the character with WASD or arrow keys (the standard keyboard keys used to move characters in PC games) and using the mouse to look around. In addition, the character controls can be a bit clunky, so the games don't reach their full potential outside of VR. No programming skills are required to create the games. The implementation is done in the browser using their visual coding language called CoBlocks. This is structured like a building block and users can choose from a wide range of assets (characters, objects, flora and fauna). These are then dragged and dropped into the game world. Using a kind of frame, users can assign different movements and texts to characters and objects. Nested structures can be created within these frames, allowing for complex processes. Nevertheless, the entry level is kept low.

However, there are a few points that the UEmbed project wants to approach differently. CoSpaces Edu is not available for free, so it is only accessible to people and projects with sufficient budget. This excludes certain groups of people and is not accessible to everyone. Although there is a free version, many features are not available and users are generally more limited in their options than in the paid version. Projects are not stored locally, but only online on the CoSpaces Edu platform. If the operating company DelighteX GmbH decides to block projects or goes bankrupt, it is possible that users' work will no longer be available. As a result, users put their projects in a certain dependency and do not have full control over their games. In addition, although the graphics are 3D, they are low-poly, simple, and cartoonish. For example, characters often have disproportionately large eyes, which makes them look cute. Many textures and assets are quite rough. This style of art does not lend itself well to certain subjects. Serious topics such as depression, war or crime seem out of place in such a scenario. This is why CoSpaces Edu is aimed more at a younger audience, while UEmbed wants to serve an adult audience as well.

 ISBN: 978-1-68558-084-1

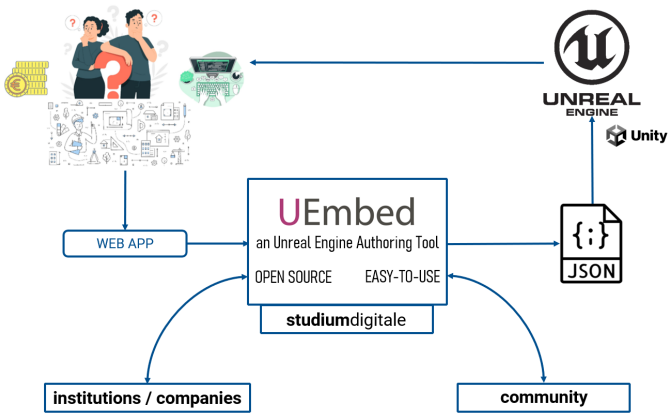

Figure 1. Overview of UEmbed: Quick, cost-effective Serious Game creation with community support.

C. Unreal Marketplace

UEmbed also allows the use of the Unreal Marketplace, an online store from Epic Games, where countless assets are offered. Many of them, such as 3D models, textures and sounds, are free and can be used by users in their games. There are also templates available, which provide a specific type of gameplay pre-programmed with all the necessary codes with placeholders. These placeholders can be replaced with any other asset of your choice. The template can also be modified at will. However, creating such assets is time consuming and requires a certain amount of creativity. It also requires programming skills, making it too cumbersome for the target audience to create such assets themselves. Therefore it is a great advantage that these assets are already created in the Unreal Engine and partly available for free. Of course, there is also the option to purchase additional assets if the budget allows. In addition to the official store, there are many other websites that provide templates or assets, often for free as well.

IV. The Approach

A. Design

The main drive behind this project is to allow everyone to participate in the game development process, regardless of their technical expertise. The workflow of creating an immersive game should be made more accessible not only to those with limited knowledge, but also to those with different accessibility.

The serious game ”Lola’s First Semester” was a visual novel-style game with 2D graphics mixed with mini-games. You might ask why we don’t want to make a 2D game anymore. We are leaning towards 3D because we want to give our players more immersion. According to Slater et al. (2013) [6] immersion is the description of a technology and evolves to the extent to which an individual is provided with an inclusive, comprehensive, surrounding, and vivid representation. Therefore, systems with high levels of immersion increase the perception of presence [7]. Presence, in turn, represents the potential psychological and behavioral response to immersion. Moreira et al. (2016) conclude in their pilot experiment that the 3D experience promotes high levels of presence perception compared to the 2D experience [7].

From here on, we will refer to people who work with the authoring tool, design and plan games as ”the user”, and people who play those games as ”the player”.

There are two points of accessibility provided with this project. Since the UEmbed Authoring Tool is browser based, it allows the user to work with it on any operating system of their choice, be it Windows, MacOS or Linux. All these operating systems are also supported by the Unreal Engine. Someone with a disability can also technically work with the authoring tool. For example, a visually impaired person could use a screen reader to navigate and use the tool. From the player’s point of view, they will be able to play an immersive and modern looking serious game in 3D or VR. The deployment of the non-VR serious game should be done in such a way that anyone with hardware and an internet connection can play it.

The user should be able to create both exterior and interior 3D environments. Buildings, nature and non-playable characters (NPC) can be placed in the exterior environment. The player would be able to explore the 3D environment. If the environment is indoor, the user can define rooms with doors on a floor and place objects in them with NPCs. The user could decide to either give the players the freedom to walk around the environments or the user could decide that the players should be on-rails. On-rails means that the players will only be able to experience the gameplay as the user has designed it to guide the players sequentially from start to finish. The user can also decide what kind of movement options are available. Since modern console games are played with a controller, Unreal Engine already has this functionality built in. But there is also the issue of accessibility. Users can select the controls to be used with a controller to manually choose the direction in which a character can move. They can also select the option where there is a list of points of interest and the player can select one by mouse click or touch (if on mobile) and the character will automatically move to the selected point of interest. The last option is very necessary not only to include players with disabilities, but also people in general who have never played a game before. Or those for whom controlling a character with a modern controller or WASD keys is too difficult.

When it comes to interaction, the player can walk up to an NPC and start talking to them. While talking to an NPC, the player will have choosable dialogues that will trigger different dialogues depending on the choice. NPCs can also give the player quests to complete in order to advance in the game. If necessary, the NPCs can also accompany the player to the place where they can start, continue, or finish their quest. The world can also contain interactive objects that can be pushed, pulled, triggered, or placed in the player’s backpack. The user can interact with these elements to create an immersive experience for the player.

If not decided before, the user can convert his game into a Virtual Reality (VR) game for more immersion at any time during the production phase or even afterwards. The VR version will get rid of the controls for the game controller to navigate, as motion sickness is one of the big problems

 ISBN: 978-1-68558-084-1

that could occur and a lot of tinkering with the settings is needed to solve it. UEmbed will include the jump navigation for the VR version. If the player has a VR controller in one hand, he can move it around and it will show a point on the ground in that direction where he can jump to in the game by clicking a button. This is not possible unless the controller is pointed at an object or NPC. The player can point and click on an object to interact with it, or the same action can trigger a conversation with the NPC. For some players with disabilities, jump navigation can be a problem. Therefore, we will take the same approach as the non-VR version and implement a clickable menu with all points of interest listed. This way the player can select the point of interest and jump directly to it.

For our part of the project, which is programming the authoring tool and preparing the Unreal project, the authoring tool will be done as a web application. This method of delivery also allows those with accessibility issues to use this tool with assistive technologies such as NVDA.

B. Technical

The result of this work should be something that is indistinguishable from the work of someone who would develop a game natively with a game engine, or even better in some cases. The products developed with this project should be able to be showcased and released to the intended audience. We also need to make sure that the exported games can be played by anyone, despite their hardware limitations.

When the decision came to choose the game engine, we looked at the most popular game engines Unity and Unreal Engine according to steamdb, a website that detects technologies used in games [8]. This site goes through each of the games available on the digital game store called Steam [9] and lists all the software used to develop them. Any search for the most popular game engines will always show these two at the top of the list. We made sure to include only those engines that were available for free, so that access to the engine could be guaranteed.

Godot was also on the list to look at. While it is lightweight and good for smaller projects, it is unfortunately not as feature rich as Unity or Unreal. A more feature-heavy approach was needed for this project, and Godot is still a new engine in development.

The basic functionalities required by Unity and Unreal are similar, but there was only one element that argued more strongly for the Unreal engine. It was the availability of usable 3D assets, which Unreal Engine provides with Quixel Megascans. These assets (3D models of building walls, trees, people, textures, etc.) can always be implemented in projects for free only when used with the Unreal Engine, which takes care of collecting quality assets with distributable licenses. This gave us more time to focus on the development side.

One question that came up was: Is the process we are following necessary? The mere existence of templates, which are pre-programmed settings, code and workflows, would seem to make the idea obsolete. Sure, there are plenty of templates available on the Unreal Marketplace (a store for developers to buy and sell assets or templates), but all the predetermined workflows for a specific game genre make them not very flexible to use. The big problem with these templates, as useful as they may be, is the necessity to have a decent understanding of the Unreal Engine workflow to make even the smallest changes to the template, let alone completely modify it according to the user's plans. This brings back all the problems of time, knowledge and budget.

The pain point that should be addressed with this approach: Having a system based on known elements that users can easily interact with. This would require splitting the project into two separate software entities: The Unreal Engine project, with all the functionality implemented in advance, and a browser-based application that the user interacts with to plan and design the game. These two systems will communicate with each other using Json and Ini files.

The core aspect of immersion is to infuse design and technology. Players are placed in a realistic 3D environment where they interact with NPCs. To increase immersion, the interaction between the non-playable characters and the player should be as close to real life as the game can simulate. Therefore, Unreal Engine's Metahuman technology was used for these characters. Metahumans are pre-configured lifelike digital avatars that are ready to use with the Unreal Engine. The user can go to their website and create their own avatars, import them into the Unreal Engine and implement them into the environment with animations. Since this process is very time consuming and difficult for users, we would provide some pre-configured metahumans for users to implement immediately. These characters also have the pre-configurations for lip-syncing. Their dialogues will be displayed as subtitles. If the dialog is recorded beforehand and the voice recordings are added in the authoring tool, they can be heard by these characters with basic lip-syncing. This will give the player the feeling of a natural conversation.

Unreal Engine also provides a set of 3D assets with its Quixel Megascans that can be used free of charge. Each asset on this platform is a 3D scan of a real object and is of very high quality. Using assets from this platform results in UEmbed having graphical fidelity on par with modern video games. In addition, Unreal Engine 5.1 has the Nanite functionality, which can render very high quality assets without much impact on performance compared to previous rendering methods.

However, the existing hardware must be taken into consideration. Not everyone will have the latest hardware to play the developed serious game. The assets used by Metahuman or Quixel can all be scaled down via in-game settings, so that the game can also run on low spec hardware. But since even the lowest spec hardware can bring a lot of problems, a solution has to be found so that everyone can enjoy the game without any problems. One solution is the pixel streaming of the Unreal Engine. When the game is ready, it can be placed on a Pixel Streaming server. Whenever someone wants to play a serious game, all they have to do is open their browser, enter the website, and the game can be played right there on their mobile or PC without any graphical downgrading. The game is running on the server, taking input from the device, sending it back, and calculating the new frames to display on the device. Since this happens in milliseconds, the player will

 ISBN: 978-1-68558-084-1

Figure 2. Top: Map created in browser-based authoring tool that exports Json. Bottom: Environment automatically recreated in Unreal Engine 5.1 after importing this Json file.

hardly notice anything. On the server side, an instance of the game is started for each player accessing the site. Therefore, it is recommended to limit the access to keep the budget low.

The collaborative development aspect is also something this project strives for. As resources for development in general is something everyone struggles with, it becomes a necessity that we try to incorporate any further development done with this project. Therefore, this project will be available as an open source GitHub project to the extent of our development work, as Unreal Engine is owned by Epic Games and needs to be kept separate. The goal is for everyone to pitch in to help this project move forward and reap the benefits.

It is imaginable that in addition to developing serious games with the authoring tool in this project, other professionals from technical departments will use the underlying scripts and functionalities to develop their own work. This collaboration will benefit everyone involved, from the users to the players. An overview of this idea is shown in Figure 1.

V. Workflow of UEmbed

For the first phase, the user must first download and install the latest version of the Unreal Engine (which is 5.1 at the time of this writing). Then download and extract the UEmbed zip file. This will contain a UEmbed Project folder and a UEmbed App folder.

The user should open the app and two options will be available: either create a new project or load an existing one. If the user chooses to create a new project, a guided process will begin to set up the project properties. During this process, the user will be asked definitive questions about the game mode (first person, third person or VR), navigation style (full 3D movement or select and click to automatically walk to the goal) and graphics (realistic or stylized). An ini file containing all the information is generated and implemented in the UEmbed project folder.

The user can now start planning the game. The planning page has five sections: Maps, Characters, Dialog, Scenes and Sequences. Each of these sections contains specific elements that can be added or deleted. First, the user should create the characters by choosing the name and selecting a predefined model. Then a map should be created, where you can specify how many buildings, trees and which of the already created characters should be added. Dialogs, which will be a conversation with choices between the user and the character, can be created separately without being assigned to a character. When everything is done, the user can create a scene that contains the map and the dialogues that will be assigned to the existing characters on that map. In this way, different scenes can be created with different combinations of maps, characters, and dialog.

All this can be brought together in the Sequence section. All created scenes can now be added in the desired order. Placeholder scenes can be placed between the added scenes so that any custom gameplay, such as mini-games, can be manually developed there.

When everything is ready, the user can click Export and each of the sections will save its elements as individual Json files in their respective folders. All the json files will have their associations to each other included via IDs. The next step would be to open the Unreal Engine project. The user can follow the documentation provided to locate each of the visual scripts for maps, characters, dialog, scenes, and sequences needed to build the world, and click on their Re-Import buttons to import the previously created Json files. Now the user can play the game and see his plans visualized. Once everything is in place, the game can be built and distributed. The example of map creation is shown in Figure 2.

Changes to the game will always be needed during the production phase. Therefore, the user can load the previously created project, make changes to it, and rebuild the game using the same process mentioned above.

VI. Conclusion and Future Work

The growing demand for Serious Games leads those working in education to look for ways to participate in this field. However, the lack of technical knowledge hinders them to communicate their approach in the context of serious games. Of course, they could start by learning game development, but in practice it is a daunting task to start even with coding skills, as each engine is filled with a large set of functionalities. Also,

 ISBN: 978-1-68558-084-1

hiring employees or even external help is limited by limited financial resources. Therefore, with this proof of concept, we are on our way to democratize the whole process of serious game development. An authoring tool that will give access to everyone who wants to present their stories in a 3D space with modern graphical fidelity.

Further work is needed in this project on the user interface (UI) and user experience (UX). It needs to be ensured that users can work with an authoring tool that is easy to understand. The immersion can also be deepened if the NPCs used here could be imbued with character AI that has certain specific knowledge and associations with other NPCs. For a more natural conversation, the NPCs could be talked to via speech recognition, with the results of the speech-to-text being fed to their internal chatbot-like AI, which will then respond to you via text-to-speech. An automated lip-sync process will help sell the immersion more.

Targeting the right hardware should also be a focus, as there are so many configurations already out there. If needed, there could be different sets of preconfigured UEmbed projects for different hardware, for example a UEmbed project just for virtual reality. This wouldn't change the authoring tool, but it would give users a faster starting point instead of having to deal with configurations themselves.

A thought process from a game design perspective will be helpful to think of use cases that are commonly used. Implementing more of them will help the community as a whole, and it will also attract more interest. If the community pulls it all together and expands this project on its own merits, it will not only expand the feature set, it will also expand the boundaries of serious games much more.

REFERENCES

[1] R. Dörner, S. Göbel, W. Effelsberg, and J. Wiemeyer "Serious Games – Foundations , Concepts and Practice". Springer International Publishing, Cham, 2016. isbn: 9783319406114. pp. 25

[2] Lolas First Semester. Last visited: 07.03.2023. [Online] Available: https://lolaserstessemester.sd.uni-frankfurt.de/

[3] L. Rustemeier, S. Voß-Nakkour, S. Mateen, and I. Hossain (2021). "Creation and Future Development Process of a Serious Game: Raising Awareness of (Visual) Impairments". In: Fletcher, B., Ma, M., Göbel, S., Baalsrud Hauge, J., Marsh, T. (eds) Serious Games. JCSG 2021. Lecture Notes in Computer Science, vol 12945. Springer, Cham. pp. 3-5

[4] E. Shliakhovchuk and A. M. Garcia (2020), "Intercultural Perspective on Impact of Video Games on Players: Insights from a Systematic Review of Recent Literature", In: Educ. Sci.: Theory Pract., vol. 20, no. 1, pp. 40–58.

[5] F. Laamarti, M. Eid, and A. E. Saddik. "An overview of serious games." In: International Journal of Computer Games Technology 2014 (2014). issn: 16877055.

[6] M. Slater, A. Steed, and M. Usoh (2013). "Being there together: Experiments on presence in virtual environments" (1990s). Technical Report, Department of Computer Science, University College London, UK.

[7] P. Moreira, E. C. de Oliveira, R. Tori. "Impact of Immersive Technology Applied in Computer Graphics Learning". In Proceedings of the 15th Brazilian Symposium on Computers in Education, Sao Paulo, Brazil,4–7 October 2016; pp. 410–419.

[8] SteamDB: What are games built with and what technologies do they use? Last visited: 07.03.2023. [Online] Available: https://steamdb.info/tech/

[9] Steam: A digital game store. Last visited: 07.03.2023. [Online] Available: https://store.steampowered.com/

 ISBN: 978-1-68558-084-1