VIRTUAL EXHIBITIONS DURING A PANDEMIC – A REAL-TIME ONLINE EXPO WITH A FICTIONAL INTERIOR

Stefan Stavrev

Abstract. *In this paper we present a novel approach for organizing an online virtual expo. The virtual representation allows visitors to navigate the environment in a manner closer to reality. We use modern real-time visualization tools such as game engines to present and interact with exhibitor's content. Unlike other researchers, our work uses a fictional interior that does not exist in the physical world. That requirement allowed us the creativity to not restrain to the physical limitations of actual halls and buildings.*

Key words: Serious games, Virtual Reality, 3D modeling.

1. Introduction

The Covid-19 pandemic has been a challenge for all of mankind. With major restriction, businesses had to find new ways of operating. That includes participating in events such as expos and conventions. Many online meeting platforms – such as Google meet [1], Microsoft teams [2], zoom [3] helped organize online courses for planned and ad hoc events. However, this traditional approach is not always the best one nor it works in all cases. Particular examples of the latter are international exhibitions. On the one hand, exhibitors need to represent their products in services in such a way that is appealing to customers. On the other hand, visitors need to freely roam across exhibition halls, interact with exhibitors, attend conventions and presentations, collect promotional materials, and possibly ask questions. And due to the COVID-19 restriction, all activities need to be performed in a virtual online environment.

Traditional methods of organizing online events include creating an event website, where visitors and exhibitors register. Each exhibitor creates an account, where they upload their promotional and informational media content. Conference lectures and presentations during the expo are typically conducted via an integrated meeting platform or a streaming service. Our proposition is to build upon the traditional and expand virtual expo organization with real-time visualization mechanisms of 3-dimentional (3D) media content. In order to achieve those goals, we propose to use modern game engines rendering capabilities combined with well-known web-development tools and methods.

2. Previous work

Online events are traditionally organized with the help of web-based tools and methods. Those include: text and multimedia rich websites and video streaming services. The websites themselves offer limited options for interactive content: 2-dimentional image maps with hot interactive areas for users to click, embedded street maps, provided by third parties (such as Google street view), information, displayed in HTML tags with CSS styles, and some user interactions with Javascript. In addition, there are no or limited use of technologies for 3D rendering, such as WebGL. Video streaming services, on the other hand can add a certain interactive element – as a lecturer speaks in real-time, a visitor watching the stream can ask questions to the presenter without delay. However, such streaming services are more suitable for conferences, and not as much for exhibitions. Visitors attend exhibitions to meet with product manufacturers. Exhibitors, on the other hand, want to show their latest products and services in the most representative way. In times of a pandemic, where physical interactions are limited, the virtual 3-dimentional representations of those goods and services are the next best thing.

Game engines are such a mature 3D visualization technology. In addition to pure entertainment, they have been used successfully in different context. For instance, some researchers [4] have created a simulation of underwater cables to help them prepare for the installation of the real ones. Other authors [5, 6] propose the use of game engines in an educational content and propose that the ideal learning environment consist of proportional use of virtual and physical elements [5]. There are examples of 3D virtual environments that focus on interactions with the users [7]. More in-depth insight on the creation of online virtual museums is presented by Alin Moldoveanu et al. [8].

There are several technical approaches of representing a 3D virtual expo or museum [9]. One approach is to use 3D modeling tools, such as computer aided design (CAD) software – ArchiCad, Maya, Blender, 3DS

Max, etc. Another approach is to use a range-based modeling. The third option is called image-based modeling.

Despite the 3 possible approaches, our choice was restricted by the fact that the desired expo did not exist in the physical world. That restriction excluded range-based modeling and classical image-based modeling due to the absence of a physical environment to shoot. We were left with 2 options: 3D modeling or computer-generated image-based (CGI) modeling. We went for the former approach for several reasons. First, for CGI – based modeling we had to create the 3D models of stands, people, equipment, etc. anyway and render still images using a renderer (such as V-ray, Octane, Lumion). That seemed like double work since we can use the 3D models directly in a 3D environment and visualize them in real-time using a game engine. For CGI-based modeling we had to first map the rendered images to a cubemap / spheremap and only after that create the actual expo space. The CGI-based modeling had another drawback – it restricts the use of active and interactive content elements. That means no real-time animations of rigged characters walking around the space, limited interactive video elements and limited use of dynamic text for banners and advertisements for the exhibitors' stands. Although, the CGI – based approach did have some advantages. For instance, loading images instead of 3D models in a web-based environment is computationally cheaper, even with high-resolution image maps. However, light reflection realism, real-time interactions and animations tilted the scales to use 3D modeling approach.

3. Implementation

For our implementation, we use the game engine software Unreal Engine 4 [10]. As we saw, game engines have a history of being applied successfully in a non-entertainment, non-gaming environment. Each exhibitor is presented with a choice they can pick up a 3D kiosk from: single stand, island stand or a corner stand (Figure 1). Each 3D model has several placeholders (Figure 5), where exhibitors' content – such as video, images and text can be embedded. The content can be managed via a web interface where each exhibitor has created a profile. Through the web interface, basic create-read-update-delete (CRUD) operations can be performed. Content can be uploaded, reviewed, modified or removed from the database. If no content is available or is removed from the user profile, a default placeholder is used in the expo scene. However, leaving empty placeholders and showing default content in the expo stands is not recommended. That degrades the aesthetic of the stand and the expo, as well.

The virtual expo is divided into several sections. Each room is a separate 3D interior scene, represented as a separate in-game level. That division is necessary in order to minimize content loading times over the network (see Figure 2 in which the green arrows represent request queries and the blue arrows – response results, rendered in the visitor's web browser). For instance, a regular entry room costs about 120 MBs of resource loading over the network. Transition between levels is achieved by using UE4's level streaming [11]. Unreal requires us to have a persistent level – that is a "master" level which governs which room or interior will be streamed in and out.

Figure 1. Example of a corner stand

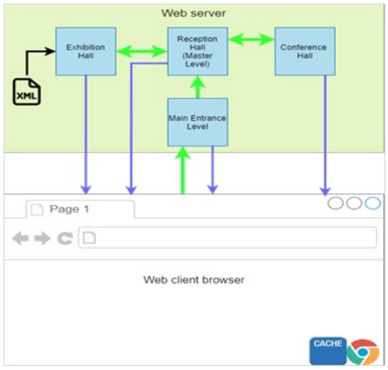

Figure 2. Client-server architecture

We use the classical client-server approach. The digital expo – with all its models and resources, is located on a central web server. Each visitor acts as a client – connects to the URL of the expo and enters the main hall. During that initial connection, the client's web browser downloads the digital content of that section. Even with level streaming, initial loading times are high. However, once loaded in the client's browser, the content is cashed locally (Figure 2). That allows visitors to only load 3D content on their first visit to the web server. There is an exception to this rule – when part of the content gets updated, all visitors have to re-download the updated content from the server. In the future, we plan to implement a "delta mechanism" – to only download the different content once it is available to the web client.

The expo interior is divided into several sections. Each section is represented as a separate in-game level. When visitors first enter the expo via a URL, they see the exterior of the expo building. For effective use of resources, the view camera and navigation are restricted to the main building entrance. 3D decorations – such as trees, benches and moving people are optimized in this scene. From there, visitors have the option to enter the main hall, called reception hall.

The reception acts as a persistent "master" level – it is always loaded into memory. In addition, it acts as a main entry point to several other halls of the expo complex. The reception has a 3D avatar behind a counter, acting as a greeting host to the visitors. The reception hall serves as a master level in the level streaming (see Figure 3). The avatar communicates with the visitors via a text prompt. Visitor's queries are entered in a text box and then send to an online assistant via the REST API. The answers are then related back to the user, much like an online chat.

From the reception room, there are several directions a visitor can go. One is towards a presentation hall. Before that hall, there is a dynamically changing schedule with all the lectures and presentations that will be held in the duration of the expo. The schedule board is updated via an XML file that lies on the webserver file system. The file can only be edited by an authorized organizer personnel via a web interface. In addition to schedules, the XML file also contains links to presenter resources, such as speaker's photo and a video recording of the lecture. Inside the lecture hall, we used a fixed-perspective camera that is pointed towards a virtual screen. The screen is a placeholder, where the presenter's video is being rendered (Figure 4).

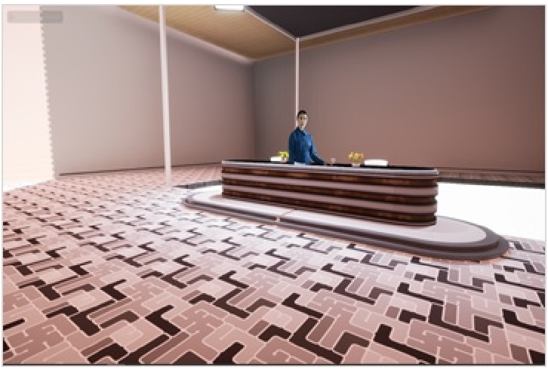

Figure 3. Reception hall

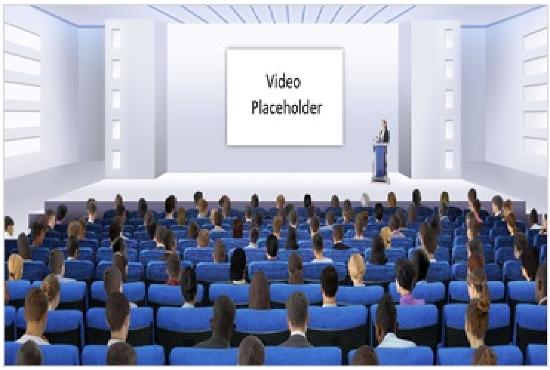

Figure 4. Conference hall

The final and most important interior space is the exhibition hall. It consists of all the stands of all the exhibitors. That hall is represented as a pre-build level with some dynamic elements. Those elements are: the banner of the stands, the text on them and the media materials on the stand. Again, as with the presentation hall, XML files are used to describe which resources should be loaded on each stand. The number of stands corresponds to the number of exhibiting companies. In future iterations we plan to expand on dynamically adding and removing exhibitor stands. In order to avoid playing all exhibitors' videos at once, we have defined an attention radius around each video. That way, the video sound will be audible only when a visitor is near a stand (Figure 5).

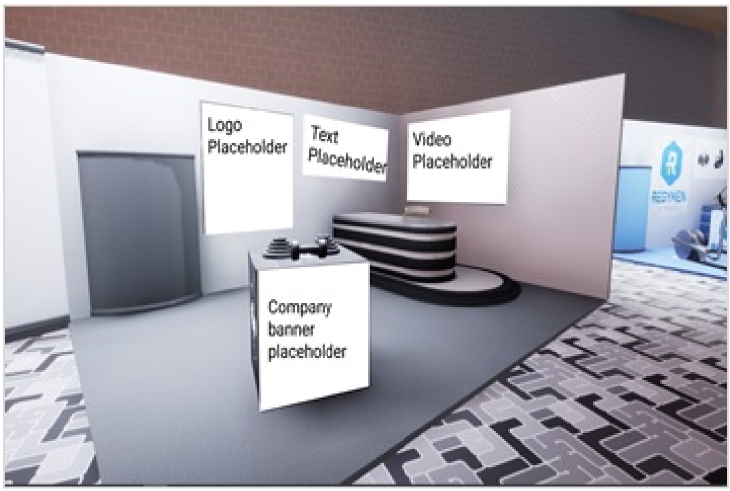

Figure 5. Exhibition hall with media placeholders for rendering exhibitors' content

4. Results

The presented approach yields several benefits. First, no actual physical space is used for reference. That unlocks the creative potential of architects and interior designers and allows seamless transitions between different expo areas. In addition, the outer dimensions of the exhibition do not have to match the interior space. The virtual expo allows a seemingly small building to house many halls, wide conference rooms, expo stands, etc.

Another benefit is the seamless transition between virtual spaces. The visitors can instantly teleport from one room to another without having to virtually walk the interior. That saves time and unloads the expo server from unnecessary resources. That is the benefit of implementing different halls and rooms as streaming levels. As we have already mentioned, 3D content is downloaded only once in the client's browser (on the initial URL access request), which is another advantage of the proposed approach.

5. Conclusion

We have presented a novel approach for creating a real-time 3D virtual expo using Unreal 4 engine and client-server architecture. The presented approach has several advantages over previous visualization techniques – mainly – more realism, rich use of media content, unrestricted use of 3D space, seamless transition between exhibition halls and several optimization techniques. The level streaming feature of Unreal engine optimizes loading times between different expo sections.

Acknowledgments

This paper is partially supported by project MU21FMI007 of the Science Fund NPD of the Paisii Hilendarski University of Plovdiv, Bulgaria.

References

[1] Google meet, retrieved on 30.08.2021 from: `https://meet.google.com/`.

[2] MS teams, retrieved on 30.08.2021 from: `https://www.microsoft.com/\allowbreaken-us/microsoft-teams/group-chat-software`.

[3] Zoom, retrieved on 30.08.2021 from: `https://zoom.us/`.

[4] O. Ganoni, R. Mukundan, R. Green, Visually realistic graphical simulation of underwater cable, *26th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG 2018)*, Plzen, Czech Republic, 28 May – 1 June, (2018), ISSN: 2464-4625, DOI: `https://doi.org/10.24132/CSRN.2018.2802.8`.

[5] A. Akhayan, and E. Teteleva, 3D virtual pedagogical studio, *ICERI2019 Proceedings*, (2019), 4221–4225, ISSN: 2340-1095, DOI: `https://doi.org/10.21125/iceri.2019.1051`.

[6] T. Cinto, et al., 3D Virtual Learning Environments: An Avatar-Based Virtual Classes Platform, *Virtual Reality in Education: Breakthroughs in Research and Practice*, edited by Information Resources Management Association, IGI Global, (2019), 193–225, ISBN-13: 9781522581796, DOI: `https://doi.org/10.4018/978-1-5225-8179-6.ch009`.

[7] X. Huang, M. Tsai and C. Huang, 3D Virtual-Reality Interaction Sys-

tem, *2019 IEEE International Conference on Consumer Electronics – Taiwan (ICCE-TW)*, (2019), 1–2, DOI: https://doi.org/10.1109/ICCE-TW.46550.2019.8991800.

[8] A. Moldoveanu, F. Moldoveanu, A. Soceanu, A. Victor, A 3D virtual museum, *UPB Scientific Bulletin*, Series C: Electrical Engineering, (2008), Vol. 70 (3), 47–58, ISSN: 1454-234x.

[9] M. Carmo, A. Cludio, 3D Virtual Exhibitions, *DJLIT – DESIDOC Journal of Library & Information Technology*, (2013), Vol. **33** (5), 222–235, DOI: https://doi.org/10.14429/djlit.33.3.4608.

[10] Unreal Engine, retrieved on 30.08.2021 from https://unrealengine.com.

[11] UE4 Level streaming, retrieved on 30.08.2021 from https://docs.unrealengine.com/4.26/en-US/BuildingWorlds/LevelStreaming/Overview/.

S. Stavrev[1,*]
[1] Paisii Hilendarski University of Plovdiv,
Faculty of Mathematics and Informatics,
236 Bulgaria Blvd., 4003 Plovdiv, Bulgaria
[*] Corresponding author: stavrev@uni-plovdiv.bg