Designing and experiencing spaces together – A low-cost VR multiplayer approach in teaching

Gehring, Maximilian; Mosler, Pascal; Rüppel, Uwe; Dokonal, Wolfgang

Published in:
Envisioning Architectural Scales in the Analogue and Virtual Representation of Architecture

DOI:
https://doi.org/10.60558/eaea16-2023-316

Publication date:
2023

Document Version:
Publisher's PDF, also known as Version of record

Document License:
CC BY

Link to publication

Citation for pulished version (APA):
Gehring, M., Mosler, P., Rüppel, U., & Dokonal, W. (2023). Designing and experiencing spaces together – A low-cost VR multiplayer approach in teaching. In A. Kreutzberg (Ed.), *Envisioning Architectural Scales in the Analogue and Virtual Representation of Architecture: Proceedings of the 16th EAEA conference* (pp. 259-266). Royal Danish Academy - Architecture, Design, Conservation. https://doi.org/10.60558/eaea16-2023-316

https://doi.org/10.60558/eaea16-2023-316

Maximilian Gehring [a], Pascal Mosler [b], Wolfgang Dokonal [c], Uwe Rüppel [d]
[a, b, d] Institute of Numerical Methods and Informatics in Civil Engineering, Technical University of Darmstadt, Darmstadt, Germany, [c] Institute of Urbanism, Technical University of Graz, Graz, Austria

Designing and experiencing spaces together – A low-cost VR multiplayer approach in teaching

Introduction

There is an increasing interest in using Virtual Reality (VR) in architectural academic teaching. Head Mounted Displays (HMD) enable users to dive into the virtual 3D model of a building. This way, clients and planners get a better understanding of a project than only using 2D plans. However, the main challenge is to create a VR environment for design purposes without having to be an expert in programming. The process should be simple enough to be manageable in the design process, but detailed enough to deeply engage the user. Users should be able to get a realistic sense of the scale and size of spaces. According to Eller (2020), there are several techniques and systems to improve this so-called immersion:

- Headphones to create a realistic acoustic environment
- VR Treadmill or similar devices to enable intuitive movement in VR
- Variable height of view in VR and a wide field of view to create a natural visual perception
- Adding realistic materials and additional visual effects to create an authentic model
- Adding objects to allow size comparison, e.g., cars, furniture or people (such as NPCs or avatars of other VR users)

There are many obstacles concerning the implementation of these technologies and systems. The offered solutions are expensive and extensive programming skills as well as complex model conversions are required to create one’s own solution. Therefore, the use of VR in the early design phase is not appealing for small companies and students – at least until now. To solve this issue, the authors of this paper have recently examined tools for low-cost VR systems using the Google Cardboard® system, which works together with an ordinary smartphone as an HMD. To improve the feeling of immersion, the original motion system provided by Google Cardboard® was replaced by devices developed by the researchers. The main aim was to replace a button on the HMD by translating the steps in the real world into the virtual world. Based on a wired solution with microswitches mounted on shoes (Dokonal, Knight, and Dengg 2016) a so called “Movement Box” with acceleration sensors and a bluetooth connection to the HMD was developed (Dokonal et al. 2021). A workflow for integrating a design model into the VR environment was formulated which can be used by architects without advanced programming and scripting skills.

Additionally, tools to alter the model in the VR environment were developed by the authors. With this toolset, a user is able to move walls and ceilings of a building model in VR and therefore change the desing and experience these changes in VR. A big disadvantage remains that only one user can dive into the VR environment. Therefore, when working in a group, the HMD needs to be passed around to discuss issues and design changes. If users could experience the vitual space simultaneously, they could collectively develop a more credible and faster understanding of the dimension than one person alone (Davila Delgado et al. 2020). Experiencing other users in the same virtual space in the form of their human-size avatar will also help to get a better feeling of the scale of the model and of the impact of design changes. In a multiplayer game, players experience the same virtual environment at the same time and see each other or their virtual avatars. This way, players interact with each other in the virtual environment and get deeply involved in the simulated environment.

This paper presents a new approach which uses the functionalities of a multiplayer game to create a collaborative VR design environment. This is achieved by using the game engine Unity and the network library Photon Fusion and combining them with the procedure mentioned above to create a VR application. The resulting multiplayer environment enables multiple users to view and alter the 3D model of a building design in VR. Since the users are connected via a server-based application, there is no need to be at the same physical location to join the VR design environment. This approach offers advantages in various applications, e.g. in teaching (design critiques) but also in practice (presentation with a customer, joint discussion of a design, etc.) (Sopher, Casakin, and Gero 2022). In November 2022 and March 2023, architecture and civil engineering students tested and refined the VR multiplayer approach in course of a workshop. The presented approach enables students to use multiplayer VR experiences in the design process.

Multiplayer basics

Multiplayer games involve two or more players who compete with each other in a shared game environment. This paper's focus is on VR multiplayer games, where players can each join the virtual world with their own device. To enable a multiplayer session, the game data must be exchanged between the individual devices via some sort of network transport. There are several libraries and frameworks for the game engine Unity, which handle the data exchange and enable the development of multiplayer games:

- Unity Transport
  This Unity package comprises a low-level networking library. It is the basis for Netcode for GameObjects and Netcode but can also be used for a custom multiplayer solution. (Unity 2022)
- Netcode for GameObjects
- This is a high-level networking library, which enables the development of multiplayer games without deeper knowledge about low-level protocols and networking frameworks. (Unity 2023a)

- Fish-Networking (Fish-Net)
  This is an open-source networking solution developed by an independent game developer. It has a feature-rich structure that allows a wide range of custom networking structures. At the same time, a high-level API also makes it possible to get started quickly without diving deeper in the matter. (First Gear Games 2023)
- RakNet 4
  RakNet is an open source, cross-platform C++ game networking engine. There exist several Unity packages developed by the community. It is used by Unity's biggest and most successful multiplayer game, Rust. (Jenkinssoftware 2023)
- Mirror
  Mirror is an open-source high-level networking library for Unity. It is based on the former UNET multiplayer networking library and still relies on the low-level Unity transport. It provides a wide range of template scripts to make coding easier. (vis2k 2023)
- Photon Fusion
  This is a high-performance state transfer Netcode SDK for Unity developed and maintained by Exit Games. The free trial allows 20 users at a time, 60 GB traffic per month and is only allowed for development purposes. (Exit Games 2023)

To implement a multiplayer game, it must be decided how the network will be structured. The so-called network topology defines how the devices relate to each other. The game can run on a dedicated server or one client can host a game, where the other clients could join. The second topology has the advantage that no dedicated server has to be maintained and updated with new models from the students. (Unity 2023b)

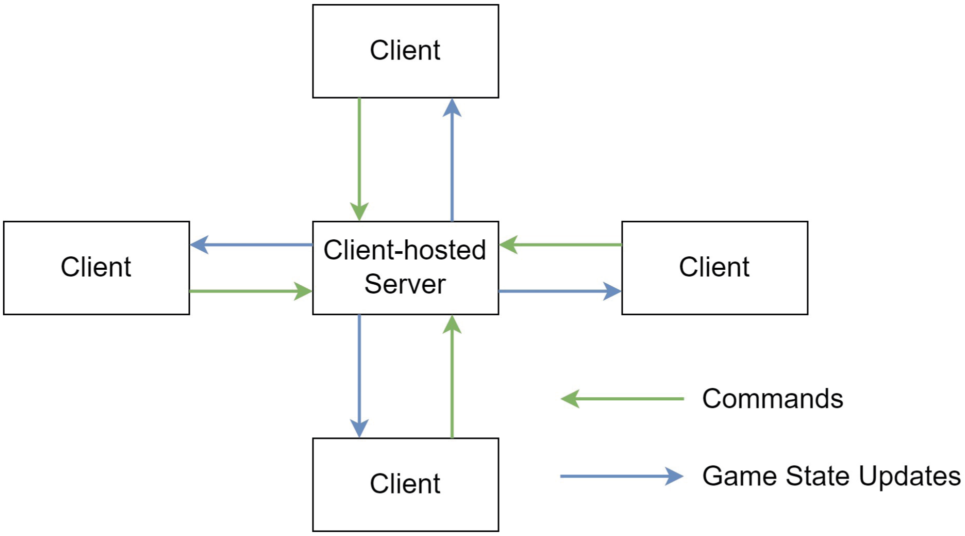

Figure 1 Choosen Multiplayer Topology
(Source: Personal Collection)

In this paper, Photon Fusion is chosen to create a VR multiplayer template project, where one client hosts the game, allowing the other players to join. The advantages of the Photon Fusion SDK are an easy to integrate network library, a good documentation, many out of the box sample projects and a wide variety of "How to" guides online. Because Photon Fusion is not open source, usage is only possible to a limited extent. This results from the disadvantage of this solution, that the game data is transferred between the devices via a server hosted by Photon.

To create user interactions in a multiplayer game, input data and the state of objects in the game have to be exchanged between the server and the clients. With the chosen approach (see Figure 1Fehler! Verweisquelle konnte nicht gefunden werden.), the clients have to send their input data to one client-hosted server. Based on the input, the server makes authorative decisions and publishes them at frequent intervals to the clients. These authorative decisions are functions which control the interaction with an object and must be implemented on the server. In addition, functions performing the interaction have to be implemented on the client side.

Workflow to create VR multiplayer

Hoon and Kehoe (Hoon and Kehoe 2003) showed the advantages of game engines in the architectural design process. Furthermore, they convinced their students of the prowess of a game engine in an elective course. With the game engine Unity, something similar can be achieved. Earlier, the authors of this paper organized workshops with students to create single player VR experiences. To go one step further, a VR multiplayer approach was developed based on the knowledge gained in these workshops.

To reduce the effort needed to set up an Unity project correctly, a template project was developed. This template project holds a minimalistic example of a floor inside a building, where the users can move the walls. Hereby, the students get a working VR multiplayer project at hand and can alter the existing project based on their own ideas. The first step is to export the 3D building model in the FBX file format. This can be done with the authoring software the students are using to create their architectural model (e.g., Rhinoceros, SketchUp, ArchiCAD, Revit etc.). The FBX file can then be imported in the Unity template project. There, the multiplayer functionalities and other game engine features can be added. In the last step, the VR application needs to be built within the game engine. A similar approach was developed by Bille et al. (2014).

Coursework

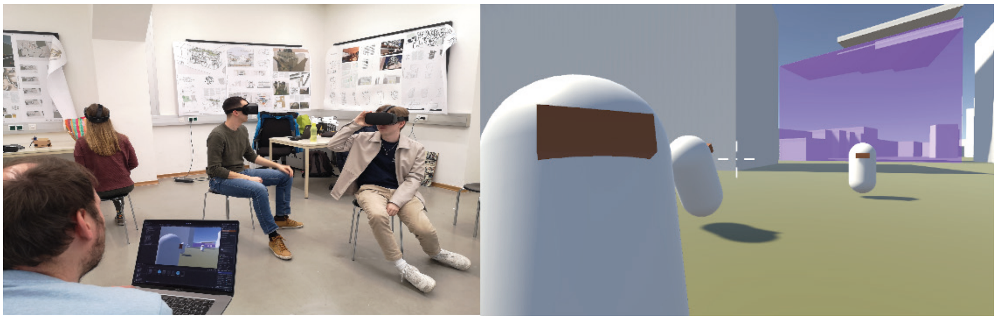

Figure 2 Students using the Project "Youth Center Graz" in the Workshop *(Source: Personal Collecton)*

In March 2023 and November 2022, two elective courses were organised, in which the students experienced how the game engine Unity supports them in collaborative modelling. Therefore, the developed multiplayer approach was presented to students in a small workshop with 5 groups. The groups had each 4 participants mixed from the disciplines computer science, civil engineering and architecture. Most of them had no prior knowledge working with Unity and only basic programming skills. Their goal was to create a VR multiplayer experience based on one of their own architectural projects. After choosing a project, they integrated in a first step their own 3D model in the provided Unity template project. In a second step, the students developed further interaction methods for their projects. With the guidance of the supervisors, it was possible to implement the ideas of the students. The following three exemplary projects stood out:

- Project „Youth Centre Graz“: In this project, a gap in the inner-city area of Graz, Austria is to be closed. The first draft only dealt with the volume of the building. In the VR multiplayer environment, the height of the building can be changed.
- Project „Max-Becker area Cologne“: This project deals with the urban restructuring of the Max-Becker area in Cologne, Germany. The chosen quarter is located in the centre of the area and has an area of approximately 100 m x 160 m. In a first design phase, it is desirable to determine the building volumes. To support this task inside the VR multiplayer environment, a user can place volumes and wall elements on a fixed grid. In addition, the storey height can be changed by extruding the placed objects.
- Project „Market place pavilion“: In this project a pavilion on an inner-city market place needs to be designed. The initial sketch consists of a canopy that provides shadows and individual cube structures that support the canopy and are used to store things for the weekly market. The following interaction options have been added to the VR environment:
    - the position of the sun can be changed
    - the cube structures can be moved around beneath the canopy
    - individual shading elements can be removed from the canopy

At the end of the workshops the students tested their projects by providing each student with the built application and presenting the interaction options live in the VR environment to the other groups (see Figure 2).

Evaluation

After the workshop, we asked all students by handing out a feedback questionnaire, how they experienced the VR multiplayer approach. This led to the following findings:

First, the students estimated the time required for the individual activities in the workshop (see Figure 3). Most of the time was used to program new interaction methods.

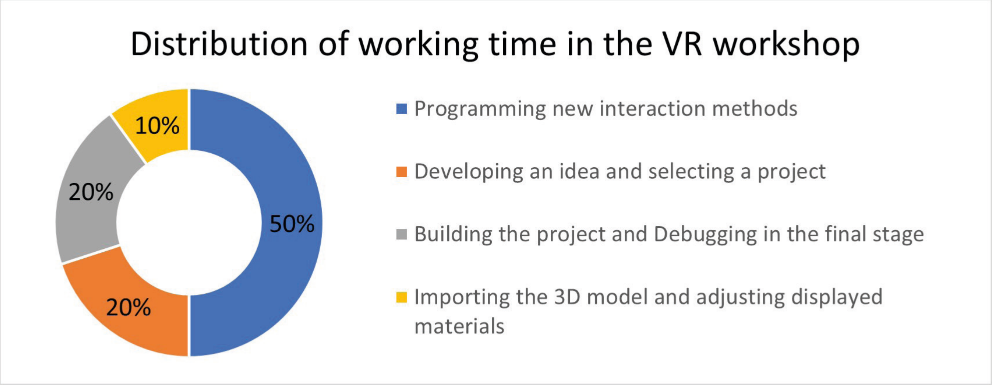

Figure 3 Students' assessment of the distribution of working time

(Source: Personal Collection)

Second, we asked the students to rate the benefits of VR and multiplayer for the tasks "design critique", "customer presentation" and "collaborative design". Furthermore, they should explain how the given approach affects their assessment. The feedback shows that the students think of the VR environment mostly as a insightful way to present a design to other people, e.g. customers. They also identified struggles when working in the VR environment with the given approach:

- The architects have little experience with VR and can therefore only poorly assess the programming effort or the possibilities of what can and cannot be implemented.
- Since CAD software already has extensive functions for implementation and visualization, there is no direct added value in using VR and multiplayer. The challenge is to provide features in the VR environment that do not exist in traditional CAD software to make VR more beneficial.

Third, the students should describe their biggest challenge in the workshop. All of them mentioned that they had major problems with understanding how new functionalities in the multiplayer environment could be implemented. As previous workshops already showed, it is easy for the students to implement new functionalities in a single player Unity project. In a single player project, one can simply use a function which regularly updates the game to collect the user input and then add feedback based on this user input, like opening a door. In a multiplayer game, the level of complexity rises drastically because of remote procedure calls. To implement a new interaction, a function has to be implemented which collects the input of the local player and sends it to the server. Subsequently a function for the server has to written, which checks if any input from remote players or the local player is at stake. Based on this input the actual feedback (e.g., make a door swing open) has to be implemented on the server. Finally, all clients have to be notified, that the door should also be opened in the games running on them.

Conclusion / Future Research

The students had difficulties adapting the multiplayer Unity project for their use case. To create new functionalities that also work with the multiplayer approach, students need a deeper understanding of the multiplayer network topology and remote procedure calls. It is nearly impossible to set up this understanding within a five-day workshop. At least with some help of the supervisors, applications useful for the design process can be programmed. Nevertheless, the students recognized the potential of the VR multiplayer approach and their interest to continue working on the topic was drawn.

Providing the developed functionalities in further workshops will help to further improve the workflow and reduce the effort for the implementation of individual projects. Then, the focus is more on testing the VR multiplayer environment in the early planning phase. Collaborative modelling of a building can be achieved with the given approach, even if not all participants are in the same physical location. Also, the visualisation of the avatars of other planners makes the model livelier and allows a better feeling of space due to size comparisons and the virtual experience of one's own comfortable talking distance inside the planned building.

In future research, the effort to adapt the template project for one's own project needs to be reduced. The aim should be, that architectural students could use this VR techniques without the need of a deeper knowledge of the game engine Unity and without advanced programming skills. Furthermore, the focus should be placed on an open-source approach, which has no limitation on the commercial usage.

Bibliography

Bille, Ross, Shamus P. Smith, Kim Maund, and Graham Brewer. 2014. '*Extending Building Information Models into Game Engines*'. In *Proceedings of the 2014 Conference on Interactive Entertainment*, 1–8. Newcastle NSW Australia: ACM. https://doi.org/10.1145/2677758.2677764.

Davila Delgado, Juan Manuel, Lukumon Oyedele, Peter Demian, and Thomas Beach. 2020. '*A Research Agenda for Augmented and Virtual Reality in Architecture, Engineering and Construction*'. *Advanced Engineering Informatics* 45 (August): 101122. https://doi.org/10.1016/j.aei.2020.101122.

Dokonal, Wolfgang, Michael Knight, and Ernst Dengg. 2016. '*VR or Not VR - No Longer a Question?*' In *Complexity & Simplicity - Proceedings of the 34th ECAADe Conference - Volume 2*, 573–79. Oulu, Finland: CUMINCAD. https://doi.org/10.52842/conf.ecaade.2016.2.573.

Dokonal, Wolfgang, Patrick Scheich, Tim-Jonathan Huyeng, and Uwe Rüppel. 2021. '*A Hard Road To Travel - Developing Tools for Low-Cost Virtual Reality (VR) Systems in the Early Design Phases*'. In *Towards a New, Configurable Architecture - Proceedings of the 39th ECAADe Conference - Volume 2*, 89–94. Novi Sad, Serbia: CUMINCAD. https://doi.org/10.52842/conf.ecaade.2021.2.089.

Eller, Christian. 2020. *Bewertung und Nutzung immersiver Methoden der Digitalen Realität für Bau- & Umweltingenieuranwendungen innerhalb der Lehre und Praxis*. Vol. 02/2020. Darmstadt: Shaker Verlag GmbH. https://tuprints.ulb.tu-darmstadt.de/12788/.

Exit Games. 2023. '*Fusion Introduction | Photon Engine*'. 2023. https://doc.photonengine.com/fusion/current/getting-started/fusion-intro.

First Gear Games. 2023. 'Introduction to Fish-Networking'. 2023. https://fish-networking.gitbook.io/docs/.

Hoon, Michael, and Michael Kehoe. 2003. '*Enhancing Architectural Communication with Gaming Engines*'. In , 349–55. Indianapolis (Indiana), USA. https://doi.org/10.52842/conf.acadia.2003.349.

Jenkinssoftware. 2023. '*RakNet - Multiplayer Game Network Engine*'. 2023. http://www.jenkinssoftware.com/index.html.

Sopher, Hadas, Hernan Casakin, and John S. Gero. 2022. '*Effect of Immersive VR on Student-Tutor Interaction in Design Crits*'. In *ECAADe Conference Proceedings 2022*, 1:123–32. KU Leuven Technology Campus Ghent, Belgium.

Unity. 2022. '*About Unity Transport | Unity Multiplayer Networking*'. 8 December 2022. https://docs-multiplayer.unity3d.com//transport/2.0.0/about.

———. 2023a. '*About Netcode for GameObjects | Unity Multiplayer Networking*'. 6 March 2023. https://docs-multiplayer.unity3d.com//netcode/current/about.

———. 2023b. '*Network Topologies | Unity Multiplayer Networking*'. 6 March 2023. https://docs-multiplayer.unity3d.com//netcode/current/terms-concepts/network-topologies.

vis2k. 2023. '*A Brief History of Mirror*'. 2023. https://mirror-networking.gitbook.io/docs/.