# Investigating the Possibilities of Using Game Engine-Based Virtual Environments in Architectural Design Studios

Tuğrul Yazar and Oğulcan Üneşi

**Abstract.** There are cognitive, technological, and pedagogical ideas and experimental studies on architectural design education in virtual environments. In the existing literature regarding this subject; Various software developed for other purposes were used, add-ons to existing software were developed depending on the scope of the studies, or software prototypes were developed and tested on various platforms such as game engines. In the distance education model, currently, there is no stand-alone software, specifically developed for the needs of architectural design studios. Similarly, there is no generally accepted pedagogical approach in this field yet. This paper, as the first step of an ongoing research project, includes identifying the transformations, challenges, and opportunities caused by the distance education model in architectural design studios. To make these determinations, alongside a literature review, semi-structured online interviews were conducted with studio instructors and students. As a result of these interviews, it is aimed to reveal the expected features of a potential virtual studio software specialized for architectural design studios. In the interviews, the prototypes of software in development were used. These prototypes were developed with the Unity game engine. In these prototypes, the users can connect to a common virtual environment via a server computer, upload their architectural project models, interact with other studio stakeholders (students, executives, jury), and make presentations. There are many features and ideas for this software that are still in the testing phase. Preliminary results show that virtual studio environments with the main characteristics of which are identified in this paper have potentials that can go beyond being just a simulation of the traditional studio. These potentials will be presented together with some basic features of the software. Traditional architectural design studios have qualities that are difficult or impossible to implement with the standard toolkit of any online platform, considering the visual and intellectual characteristics of the design, the qualities of the representational environments, and the immediate possibilities of using computational design technologies. Similarly, online design studios also have qualities that are difficult or impossible to implement within the standard pedagogical framework of any face-to-face method. The virtual environment and distance learning software developed for architectural design studios will also contribute to the field of architectural education.

**Keywords:** architectural design studio, distant learning, virtual studio software, game engines

## Introduction

New digital design and communication technologies have a significant impact on the research area of architectural education. The integration of distance education opportunities with architectural design studios has also begun to be examined and discussed within this context. These studies were previously carried out in controlled and planned experimental environments. However, in the last two years, architectural education worldwide has had to be conducted through a mandatory and extended distance education process. This uncertain and rapidly evolving process has created a broad experimental and observational environment questioning the compatibility of traditional architectural design studios with distance education models. It is believed that this environment has initiated a new momentum in distance education research within architectural education.

It is possible to observe that remote access and interaction technologies have evolved under the headings of virtual reality and augmented reality and have spread to many fields. Two of these fields are architectural design and architectural education. When examining research in these areas, it is evident that some technological infrastructures used in computer games have been utilized in architectural research. Computer games, especially in the context of the internet and related technologies that have developed in recent years, are a field where remote access and interaction are experienced and rapid developments are occurring. Technologies and platforms exist that allow numerous participants from different regions of the world to be in the same digital model seamlessly and interactively at the same time. These technologies, which are not limited to gaming and social interaction and have been used for some sports events during the pandemic, can be customized according to their purposes. Generally, it is possible to develop customized software for the needs of architectural design studios using platforms known as game engines.

Currently, there is no widely accepted or commonly used software specifically developed for the needs of architectural design studios in remote education models, nor is there a broadly accepted pedagogical approach in this field. This study aims, as the first step of an ongoing scientific research project, to identify the transformations, challenges, and opportunities brought about by the remote education model in architectural design studios. To achieve these identifications: (1) a comprehensive literature review on the subject will be conducted, and (2) semi-structured interviews will be carried out with studio instructors and students from the Faculty of Architecture at Istanbul Bilgi University to gather current online studio experiences.

A large number of remote-access virtual environment software is currently in use and new ones are being developed. However, these software solutions do not fully encompass the "customized functionalities" required for architectural design studios, which is the focus of this research. This paper will define these specialized functionalities/needs and provide insights into which components of Unity game engine technology could address these functionalities/needs. There are also interactions, representations, and simulations that may be needed by remote architectural design studios but are absent in technologies focused on computer games. Given the traditional standards of studio representation, it is considered that this issue should be investigated starting from the point of identifying needs. The integration of computational design methods offered by today's digital design technologies with the digital studio environment will also be evaluated within this framework.

## Virtual Environment Research in Architecture

In architectural education, there are technological and pedagogical ideas and experimental studies related to the use of virtual environments and distance education technologies. Below, some of these studies are evaluated based on their approaches to the subject and the results they emphasize.

In their study, Coşkun and Çağdaş (2018) explored the potential contributions of computer games to architectural education. They highlighted how digital games support a more active and dynamic learning culture compared to the homogenizing and passive learning context of traditional education. The study, which includes a method for comparing and translating the characteristics of digital games and fundamental design education, examined the topic in terms of game types, targeted skills, and the roles of facilitators. The experimental study involved an exercise process with offline and online phases following each other. In the online phase using the game "Minecraft: Education Edition," the design exercise was repeated and recorded. The study concluded that the "hybrid" use of online and offline exercises could be preferred for integrating the positive aspects of both environments.

Gül (2020) summarized the history of the use of computers, computational design, and virtual environment technologies in the field of architecture, as well as the potential future uses of these technologies. He states that with the widespread use of computers, a computationally-oriented foundation has been established in architectural production, application, and design. He notes that the introduction of affordable personal computers in the 1990s led to an increase in the use of computer-aided design programs in the design field, and from the 2000s onwards, architectural design-focused software-enabled analysis, simulation, and structural and formal form-finding methods to be carried out in a computer environment. In the context of tools used in design being decisive factors for defining the boundaries and possibilities of design, he argues that the ease of use of these tools has led to the popularity of curved and parametric surfaces. He also mentions that popular virtual environment software of the time was tested in many architecture schools and design studios. In his studies, Gül investigated the potential of multiplayer virtual environment software such as "Active Worlds" and "Second Life" in architectural design, emphasizing the importance of interaction. He asserts that to create a convincing, realistic, and tangible experience, a seamless and natural human-computer interface is required, which can be enhanced by emerging virtual reality technologies. According to Gül, the most important features of virtual environment technologies in architectural design are their ability to facilitate interaction and provide an experience-oriented environment. Based on this, he argues that virtual environment software used in architectural design should not only have data processing and computational capabilities but also focus on the experience.

Grasser et al. (2020) explore the possibilities of real-time collaborative design in architecture using their developed multiplayer and cross-platform supported application. The software, developed with the Unity game engine, utilizes augmented reality technologies on mobile devices while also being used simultaneously on computer environments with participants from different platforms. They have tested remote collaborative design scenarios with the software, which also includes text-based communication functions. Grasser et al. report that real-time collaborative design is productive and dynamic. They argue that the use of virtual environments is increasing day by day in all areas of life, and the most popular virtual environments are those that support user-generated content, with collective creativity and knowledge sharing arising from this content. They acknowledge that collective creativity and collaborative design may lead to issues regarding authorship but emphasize that the primary goal of such work methods is to produce a common output. Grasser et al. state that environments that facilitate collaboration can support multiple goals and perspectives, and in situations where real-time interaction and feedback are possible, exploration and experimentation in design are also achievable. They also mention that using multiple software due to workflow organization poses challenges.

Chien et al. (2020) proposed a workflow aimed at effectively, simply, and quickly connecting computer-aided design (CAD) software with virtual reality (VR) hardware. They note that most VR-supported virtual environments are research-focused and not easily usable by architects, and existing software suitable for architectural design is often inaccessible due to high costs. Chien et al. describe a workflow proposal that can be easily replicated, including technical details. They point out that models transferred from CAD environments to VR environments are not suitable for VR due to the format and geometric properties in which they are exported from CAD software. Chien et al. divide the methods for transitioning between environments into three categories. "Conversion" Method: This is a unidirectional method suitable for presenting models transferred from CAD to VR, but it is not appropriate for use during the design phase due to the potential loss of data. "Link" Method: This bidirectional method involves preserving data lost during conversion with special techniques and matching it through complex methods during the conversion stages, which can lead to data loss and various challenges. "System" Method: This method involves developing a VR-based CAD environment. Chien et al. argue that while this method addresses the difficulties of designing in VR and acknowledges that architects work more effectively in CAD environments, it is also not ideal. Chien et al. suggest that the most suitable use of VR technologies in architectural design is in communication and participatory design. They propose a schema to support these areas. Their suggested schema involves modifying the Rhino model with Grasshopper commands to make it suitable for use in a VR environment. Instead of designing directly in VR, their approach involves separating the variable/parametric components of the design from the rest of the model and programming them to allow manipulation in the VR environment. Interactive objects programmed in the "Unity" environment are combined with a fixed model and transferred to the "VRChat" game to be experienced in VR. Changes and design decisions are saved in "JSON" (JavaScript Object Notation) format, which is used for data transfer between most programming languages and software. These changes are read with Grasshopper commands and applied to the Rhino model. With this method, interactive game objects created as "Prefabs" in Unity can be manipulated in VR environments through operations such as altering position, orientation, and size, combined or separated, and objects programmed to include parameters and/or variations can be easily transformed in VR.

Hong et al. (2019) investigate the use of "Multi-User Virtual Environments" (MUVE) in architectural design in their research. The study focuses on the potential of these environments for remote collaborative creativity. The researchers tested the use of "Second Life" and "Groupboard" software in this context and evaluated participants' design outcomes using the "Consensual Assessment Technique" (CAT). Hong et al. argue that the environment used for creative collaboration should form the basis of communication among designers and reflect the three-dimensional form and volume, structural performance, and usability of architectural design. They suggest that overly abstract forms of representation could disrupt communication and lead to communication problems due to misinterpretation. They state that multi-user virtual environments provide an alternative to physical co-presence and that the communication tools offered by these environments support creative collaboration. The possibility of immersion enabled by these environments provides a sense of being present in the design space, and the use of "avatars" in the virtual environment strengthens the sense of presence and creates opportunities for interaction. Another effect of using avatars is the understanding of the physical and functional characteristics of the design. They also note that the presence of simultaneous and shared digital objects in the virtual environment contributes to the sense of presence and the environmental atmosphere of the virtual space. The experiments revealed that designs created in multi-user virtual environments were assessed as more "innovative" and "appropriate" compared to those made in online drawing software. The researchers believe this result is due to the use of avatars and the ability to be together in multi-user virtual environments, which strengthens communication and collaboration among designers. According to their observations, participants in these environments, which contribute to sharing and experiencing spatial knowledge, explored design decisions with the help of avatars and had the opportunity to experience physical characteristics in the digital environment. Hong et al. conclude that multi-user virtual environments are suitable for ergonomic, human-scale design projects and for analyzing design performance related to user activities in these projects. They suggest that these environments can be used in collaborative design work and collective and qualitative evaluation methods.

Sandstrom and Park (2019) developed a shape grammar game focused on spatial configurations in architectural design. The goal of the developed software is to derive rules from spatial configurations created, rather than creating spatial configurations within the context of rules. The software, developed with the Unity game engine, tracks users' movements and decision-making mechanisms and provides analyzable data about users' decision-making methods from the recorded data when the game is finished. This study serves as an example of how user statistics can be used to infer insights about decision-making mechanisms in architectural design through the recording of usage data.

Leitão et al. (2019) compare the potential of game engines in architectural design, specifically in terms of visualization capabilities, with architectural visualization engines. In their research, they compared the Unity game engine with ArchiCAD's "CineRender" visualization plugin, focusing on aspects such as navigability, the time required per visual frame, and the visual quality of the output. Leitão et al. note that the increasing capabilities of modern architectural design tools mean that increasingly complex designs can no longer be expressed using two-dimensional representation methods. They discuss the importance of three-dimensional visualization tools in architecture and address the shortcomings of three-dimensional modeling/CAD software in terms of visualization. They highlight that plugins and standalone software developed to address the visualization deficiencies of modeling software often require long durations to produce visuals, emphasizing the need for software that allows for intervention and real-time visualization. They argue that game engines, which offer real-time visualization and interaction capabilities, can meet this need but note that transferring architectural models to game engines is often problematic.

The researchers suggest that digital modeling in architecture can be divided into BIM and CAD, with CAD models being more easily transferred to game engines due to their purely visual nature compared to data-rich BIM models. The advantages of using the Unity game engine for architectural visualization include ease of integration with current technologies like VR, programmability of features such as three-dimensional sound and climate/season simulations, the ability to be used early in the design process without time loss due to real-time visualization capabilities, and providing more realistic navigation options.

In their experiments, the most notable difference was the time required per visual frame. They completed a section of video sequences of the same length in Unity in 9 minutes, compared to 470 hours with "CineRender." According to the results, game engines optimized for performance in games have lower visual quality, but the visualization quality remains consistent even during navigation, with no waiting required for visualization. The advantages of using Unity over visualization engines like "CineRender" include uninterrupted visualization, real-time texture, light, and shadow visualization, realistic interaction and navigation, VR support, and context-dependent performance/quality adjustments. Another feature mentioned but not applied in the study is wheelchair simulation in game engines.

Gül (2019) conducted a study investigating the impact of tools used in real-time collaborative design on designers' behaviors. Gül emphasizes that the tools used for representation affect designers' behaviors, noting that while various cognitive and interactive tools provide valuable support, studies addressing the effects of continuously evolving virtual environment technologies on designers' perceptions, communication, and interactions have become outdated. He points out that virtual worlds while mimicking the real world, offer a more suitable environment for design by removing physical constraints and enhancing creativity. Among these virtual worlds, he identifies "Second Life" as particularly suitable for design due to its independence from gaming mechanics, its modeling capabilities, and its adequate graphical quality.

In the study, a mobile augmented reality software developed using the Unity game engine was tested against paper and pencil, collaborative digital sketching via the internet (Groupboard), and a multi-user virtual environment software (Second Life). The tests were evaluated using protocol analysis methods, revealing that two-dimensional environments triggered discussions on behavioral and functional aspects, while three-dimensional environments stimulated discussions on structural design criteria. Another observed behavior in three-dimensional environments was the use of time focused on production and organization. The study also noted that, regardless of the technology used, being in the same physical environment promotes collective work, whereas remote collaboration tends to promote individual work. This phenomenon is attributed to individuals having their camera perspectives and the ability to move independently within virtual environments.

Another significant finding of the study is the effect of observing actions on the design process conducted in virtual environments. It was noted that design actions not transmitted in real-time in virtual environments lead to communication problems. In situations of remote collaboration without real-time action awareness, advanced planning and division of tasks are necessary. Conversely, when individuals share the same perspective—where "what you see is what I see"—planning is reported to be more successful.

Bartosh and Philip (2019) explored the potential of VR technologies as a tool in architectural education. Their research focuses on the use of VR technologies in the early stages of design for information visualization, dynamic interaction, interactive learning, and communication purposes. Bartosh and Philip argue that VR technologies represent the next step in the use of digital technologies in architecture, summarizing their advantages as synchronous presence, real-time modeling, three-dimensional data visualization, and the use of design tools within the environment.

They link the limited adoption of VR technologies in architecture to the restricted programming knowledge among architects but note that the production of affordable VR hardware has renewed interest among designers. Despite the challenges of using VR technologies typically in the final stages of design and the difficulties in transferring CAD models to VR environments, Bartosh and Philip highlight the potential benefits of VR technologies in the design phase due to their immersive and interactive features.

They suggest that programmable interaction features enable VR technologies to serve not only as representation tools but also as design and analysis tools. However, they also acknowledge that transferring models from CAD environments to VR can ease the learning curve of VR tools. They discuss the limitations of existing VR software and how these constraints can be overcome through the use of game engines. They conducted various experiments using the Unity game engine, simulating environmental data such as acoustics, daylight, and ventilation in three-dimensional environments, and programming various interactions.

The experiments revealed that using game engines emphasizes the experiential aspects of design. Participants preferred visual interfaces over voice commands within the game engine's capabilities and found that using VR environments for design allowed them to work beyond conventional assumptions.

Pienaru (2018) investigated the potential uses of game mechanics and game technologies in urban design concerning big data. Pienaru suggests that utilizing urban data (big data) that is publicly available but not typically accessible to mainstream users in a game context could benefit urban design. To explore this, Pienaru developed two different games with two groups of participants. One game was created using the "Processing" visual programming interface, and the other was developed with the Unity game engine. Pienaru states that these games provide environments that can securely simulate design decisions and enable the use of vast amounts of data through game mechanics.

Nandavar et al. (2018) proposed a bidirectional, software and hardware-independent, open-source workflow prototype between BIM software and VR environments. They argue that VR tools are not only for visualization but also enhance collaboration and communication. Their research involved examining various workflow models and current market software, detailing the shortcomings of these models. Based on their research, they presented a workflow proposal using the IFC format, which preserves metadata from BIM models, and the Unity game engine. In their developed prototype, they employed various data compiler code modules within Unity to enable bidirectional data flow, allowing manipulation of BIM objects in Unity and transferring changes back to the BIM environment. The authors emphasized the importance of data visualization in BIM and discussed the various capabilities provided by VR technologies. They noted that game engines enable graphic and performance optimization and interaction. Their prototype included features such as human-scale navigation, distance, and area measurement, object querying, transformation and deletion, visual marking, screenshot capture, and voice messaging. They highlighted that VR's 1:1 scale representation and presence effects enhance remote work, making it efficient in terms of effort, time, and resource use for large-scale projects requiring numerous experts. The researchers noted that VR enhances communication and user experience, and they plan to explore the multi-user potentials of game engines in future work.

Sorguç et al. (2017) investigated the potential uses of VR technologies in architectural education. The research was conducted within the framework of the “Digital Design Studio” courses at Middle East Technical University (METU). The researchers argued that VR technologies could create their reality rather than just mimicking the real world. They conducted their studies based on the premise that VR represents a new environment for space and experience.

According to Sorguç et al., VR experiences stimulate at least three types of learning methods, including visual, auditory, and kinesthetic. They pointed out that design studios need to be re-examined in the context of developing technologies, noting that computational design courses in architecture have become increasingly challenging. The researchers emphasized that, in such scenarios, it is necessary to teach both design and the use of related technologies. They also argued that the limits and potentials of the used environments and tools should be recognized and pushed further.

Moleta (2017) investigated the potential of dynamic environmental simulation in architectural visualization within architectural design studios. Focusing on the integration of dynamic environmental elements in virtual spaces, Moleta argued that such interactions would enhance user experience and strengthen the immersive potential of virtual environments. He observed that the increased detail level in VR environments often led users to focus more on exploring the surroundings rather than the structures themselves. In detailed and large virtual environments, users tend to navigate until they have seen everything or reached the boundaries of the virtual space, after which they may feel as though the "game" is over. Moleta noted that while photorealistic virtual environments are achievable with current technologies, they do not provide insights into the relationship between structures and dynamic environmental conditions. In his research, he initially developed software that allowed users to control environmental conditions. He found that while game engines could support dynamic interactions, real-time data control was more intriguing. Consequently, he used real-time climate data from platforms like "Yahoo Weather" and "WUnderground" in architectural visualization. Moleta found that, unlike the movement actions supported by game engines, participants observed variable environmental conditions by pausing at a point and even revisited the virtual environment on different days and times to check changes.

Black and Forwood (2017) explored the potential of using game engines in engineering simulations in their research. Their study focused on the interactive visualization of complex structural analyses on a game engine platform, emphasizing the communication benefits of real-time data visualization. To visualize the structural analysis of a moving façade, they used the Unity game engine, which offers real-time visualization capabilities. Black and Forwood noted that complex structural behaviors, like those of moving façades, cannot be effectively analyzed with static calculation methods. These scenarios often require intensive and repetitive processes with traditional methods. They highlighted that using game engines for engineering calculations allows real-time data visualization. The software developed in their study can be distributed as a compiled EXE file or even in web-based WebGL format without needing additional software or programming knowledge.

Moleta (2016) investigated the use of real-time virtual environments in digital design studios and reported on the ongoing research in this area. He argues that integrating game mechanics into virtual environments allows these spaces to be used beyond mere visualization. Although game engines have been used by architecture students for some time, Moleta notes that their application has been limited to visualization due to the challenges associated with using these engines. This has led to their use primarily in the final stages of design.

Moleta suggests that game engines could be utilized in design studios from the conceptual stage onward. He believes that game mechanics can enhance active participation in design studios. While simulation and scaled models may not fully provide a sense of presence, game engines can be used to express the experiential aspects of design. He posits that game engine technologies can bridge the gap between design and construction by allowing the experience of a design before it is built. His research indicates that students better understand human scale, dimensions, and distances through first-person perspective use and navigation. Additionally, using game mechanics to design experiences has behavioral effects on students. For instance, even students who tend to be protective of their designs and keep them hidden from others have shown a tendency to share their projects, seek feedback, and develop their projects based on that feedback within the studio context. Moleta concludes that using game engine technologies in design studios encourages students to think more user-centered and allows them to experience their designs from within rather than just observing from the outside.

Du et al. (2016) addressed the lack of multi-user VR environments in the Architecture, Engineering, and Construction (AEC) industry and developed and tested a software solution using the Unity game engine. The study focuses on communication and collaboration, suggesting that a multi-user VR environment could support project planning and scheduling in the early stages of construction. The Unity-based software includes interactive navigation, annotation, and voice communication tools. It was created by transferring and adapting BIM models to the game engine platform and compiling the software model. The researchers also explored the potential of BIM-supported, game-engine-based software for emergency evacuation simulations.

Valls et al. (2016) share their experiences with using game engines in architectural design studios. Their work focuses on the potential of game engines for simulation and gamification (first-person perspective) in architecture. They discuss the relationship between architecture and computer games, noting that this connection began with the integration of architectural elements into background designs and later became central through city-building games.

Valls et al. suggest that agent-based simulations used in city-building games could be employed as tools for urban planning and that virtual environments created with geospatial data are already utilized in geographic analyses. They also mention the use of realistic environmental simulations developed with game engines as decision-making tools in collaborative design within landscape architecture. The researchers argue that innovations like CAD software have influenced architectural form, and similarly, environmental simulations could have comparable impacts. They preferred using Unreal Engine for its lower programming requirements at the basic level but indicated that they might switch to Unity in future studies. They emphasize the importance of maintaining a short feedback loop during development to meet user requirements and avoid unnecessary features. Their work shows that using a first-person perspective and navigation, rather than the typical rotation and panning controls, enhances the sense of presence and human scale within the space. They also note the potential to collect user navigation and exploration data using game technologies and plan to incorporate this into their future research.

Yan and Liu (2007) researched developing a gamified architectural software that can be used in architectural design practice and education, focusing on sustainability and design performance simulations. Their work involves creating a connection between Building Information Modeling (BIM) software, Revit, and the Microsoft XNA game engine. The study concentrates on areas such as user profile simulations, resource management, performance analysis based on building materials, and budget management.

Although the software and versions used in their research are outdated, the methods applied, including agent-based design and analysis tools using computational techniques such as Dijkstra's shortest path algorithm, remain relevant.

The summarized literature indicates a trend where various software originally developed for different purposes has been utilized in architecture, with modifications and enhancements made depending on the scope of the studies. Prototypes have been developed on platforms such as game engines. The technologies and approaches vary, and when categorized by topics, the following list emerges:

- Utilization of Existing Virtual Environment Software in Architecture: This includes adapting and applying virtual environment tools to architectural tasks.
- Transfer of Commonly Used Architectural Data Types to Game Engines: Focuses on the integration of architectural data into game engines for various applications.
- Exploration of Interaction and Visualization Capabilities of Architectural Models in Game Engines: Investigates how game engines can enhance the interaction and visualization of architectural models.
- Use of Game Engines for Interactive Learning in Architectural Education: Examines how game engines can be used as interactive tools for architectural education.
- Application of Game Engines as Computational, Analytical, and Simulation Tools: Studies the role of game engines in performing calculations, analyses, and simulations within architecture.
- Utilization of Virtual and Augmented Reality Tools in Architecture through Game Engines: Investigates the application of VR and AR technologies in architecture facilitated by game engines.
- Development and Use of Collaborative Design Tools through Multi-User Virtual Environments: Focuses on creating and using multi-user virtual environments as collaborative design tools, often leveraging game engines.

In cases where existing software lacks certain computational, simulation, interaction, or analysis capabilities, or where multi-user features are absent, the common approach is to develop enhancements or use game engines. Game engines are preferred due to their ease of use and the flexibility they offer for developing prototypes and adding functionalities.

## Online Architectural Design Studio

### Interview I (29.6.2021)

The purpose of this discussion was to gather initial feedback from faculty members about the mandatory remote studio experiences conducted during the 2020-2021 academic year. The discussion involved six faculty members representing four classes from the Faculty of Architecture and lasted approximately 1.5 hours. The evaluations during this discussion covered the following topics, with the concepts particularly focused on in this research highlighted in bold:

- **Scale Perception Issues:** It was observed that some students working on screens experienced difficulties with scale perception. Additionally, students tended to undertake larger-scale projects compared to previous years.

- **Loss of Studio Identity:** The traditional culture of gathering and discussing within a physical space was missing, affecting the identity of the studio.
- **Lack of Extracurricular Activities and Social Interaction:** The absence of extracurricular activities and socialization led to difficulties for students who were unable to learn from one another.
- **Communication Problems:** Changes in communication methods led to difficulties in reaching students. The lack of gestures and facial expressions created unexpected challenges.

- **Camera Usage:** It was noted that having the camera on in online studio environments provided a partial sense of presence, whereas turning off the camera resulted in a complete loss of visual connection and the feeling of being present.
- **Understanding Project Areas:** Students had to rely on secondary sources to familiarize themselves with project areas, which affected their scale awareness.
- **Screen-Based Presentation Formats:** Online presentations enforced a screen-based approach, which often resulted in a sequential presentation format rather than the simultaneous review of posters typical of traditional jury settings. Tools like Miro were not deemed sufficiently engaging.
- **Accumulation of Work:** Issues related to accumulating work were discussed. The feeling of starting from scratch each week in digital presentations and the inability to accumulate feedback was noted.
- **Adaptation and Variation:** The process highlighted the emergence of different approaches and adaptations to the new situation in studios.
- **Technical Skills in Digital Modeling:** Students had to improve their technical skills in digital modeling. The digital environment suited some students' working methods better, leading to fewer difficulties or even improved performance for those students.
- **Inter-Studio Communication:** It was noted that communication between studios needed to be addressed separately.
- **Importance of Research Projects:** The importance of the research topic and purpose was emphasized, and it was noted that such studies are now more necessary than ever.

These observations provide insight into the challenges and adjustments required for remote architectural design studios and highlight areas for further research and improvement.

### Interview II (7.7.2021)

The second focus group discussion was conducted with a total of 5 volunteer students from different classes and lasted 2.5 hours. The discussion primarily addressed the challenges and opportunities experienced by students during the process, and two feedback points deemed significant for this research were identified:

- **Increased Importance of 3D Modeling:** It was observed that the potential impact and importance of three-dimensional modeling in design and presentation processes increased during remote studio education.
- **Concentration Issues and Lack of Collaborative Environment:** Students discussed concentration problems during the online period and the shortcomings resulting from the absence of collaborative studio work. It was mentioned that students were unable to concentrate on the lesson if they did not have their cameras on.

### Interview III (17.9.2021)

The second focus group discussion with studio coordinators aimed to reassess the feedback from the first meeting and to showcase the early version of a developing software (Figure 1). The meeting lasted 1 hour and 10 minutes and involved 4 faculty members, with two being new and two having participated in the first discussion. After an introductory briefing about the project, Oğulcan Üneşi presented an early version of the Online Virtual Studio (OVS) software, which is being developed using the Unity game engine. This software aims to move studio presentations to a multi-user virtual environment, providing a 3D space where models exported from CAD software are loaded. Therefore, the focus is on presentation rather than design. The feedback received from the faculty members during this discussion is summarized below:

- **Importance of Parallel Projections:** Beyond capturing perspectives with escape views inside the model, the importance of parallel projections, particularly needed in architectural studios, was emphasized. It was discussed that one possible counterpart in OVS could be a key plan (analogous to a mini-map in digital games).
- **Analytical Potential:** It was suggested that analyzing the design beyond just presenting it in the virtual environment could be highly educational. However, it was noted that advanced analyses, such as acoustic performance, would be difficult to compete with professional software, so initially, the goal should be to aim for general informational analyses in the studio environment.
- **Potential for Design Modification and Discussion:** Although OVS is primarily intended for presentation, its potential for design modification, discussion, and debate was highlighted. Even without 3D modeling, the importance of tools for note-taking and sketching for studio communication was emphasized.

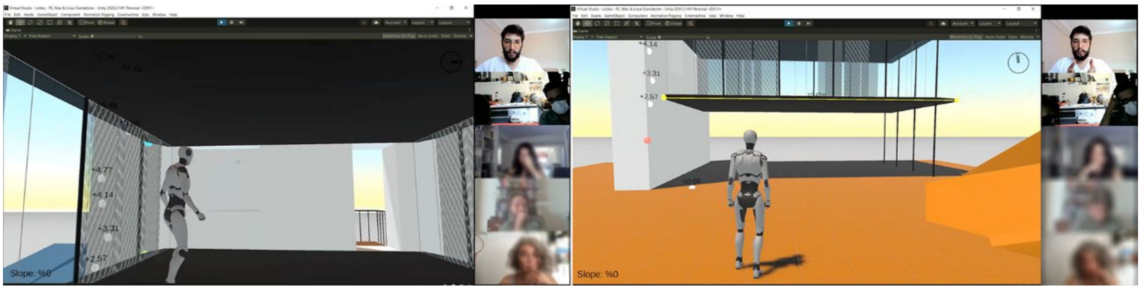

**Figure 1:** Third meeting. The presentation of the OVS, which can only be run offline at this stage, was shared via screen sharing on Zoom.

## Interview IV (3.3.2022)

The third meeting with studio facilitators aimed to allow faculty members to experience the OVS (Online Virtual Studio) from a remote, multi-user perspective. The OVS session lasted 1.5 hours, with 3 faculty members actively participating and 2 others as passive observers and commentators. This experience revealed a key difference: the virtual studio environment is not merely about interaction within a digital model but has evolved into a distinct studio design problem encompassing various scales, presentation techniques, and forms of interaction.

Discussions following the trial use of OVS with faculty members highlighted that different stages of a studio have different needs, and thus, the virtual environment can be reinterpreted to address these needs.

The virtual test studio used included a 3D model of Le Corbusier's Villa Savoye, virtual boards displaying project drawings, and models of some of the architect's furniture placed around the boards. According to this example studio setup, users could shrink to a smaller scale by stepping onto the central model (Figure 2-right), allowing them to navigate within the model. Conversely, when they jumped off the table, they would enlarge to a scale appropriate for viewing the boards and furniture (Figure 2-left).

Although this basic experience design might be ordinary for digital game design technologies, it was new for the faculty members, and the potential benefits for both facilitators and students in architectural studios were discussed. It was understood that the sequential digital presentation in remote education could be replaced by a more holistic, free-form, but structured interaction. The virtual environment's ability to maintain project stages on a server, independent of time and space, could reduce the feeling of losing progress and starting from scratch. The discussion also highlighted the need for the OVS environment to transition from a "game-like" appearance to a more "architectural" one. Consequently, potential toolsets for development were discussed, some of which are listed and explained in the Results section of this document. It was also noted that, with proper design, the software could be useful not only in studios but in other classes as well, though this topic is outside the scope of this research.

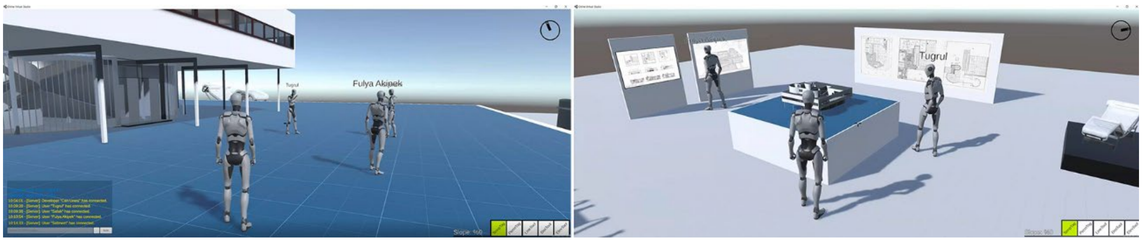


**Figure 2:** Fourth Meeting. The first remote access trial of the OVS software was conducted with studio instructors.

## Conclusions and Future Study

Traditional architectural design studios possess qualities that make them difficult or impossible to conduct with the standard toolsets available on any remote access platform, considering the visual and cognitive aspects of design, the qualities that representational environments must have, and the design and analysis capabilities offered by computational technologies. Similarly, remote digital architectural design studios also have qualities that are difficult or impossible to conduct within the standard pedagogical frameworks of any face-to-face method, considering the visual and cognitive aspects of design, the qualities that representational environments must have, and the possibilities of using computational technologies. Below, the initial functions obtained through a literature review, interviews, and evaluation of standard techniques in gaming technologies are described. These functions will be examined under 6 groups:

### Character

**Avatar:** To enable users to perceive the virtual environment relative to the human scale and interact with each other within that environment, avatars in human form and scale can be used. Therefore, it is anticipated that avatars will represent users in the virtual environment. The control scheme needs to be either first-person or third-person perspective, allowing users to keep the avatars' scales within their perspective. To animate the avatars' movements or interactions, "rigging" is necessary. Different avatars or customizable appearances are crucial for users to differentiate and recognize each other. It is anticipated that identical avatars that cannot be distinguished may create issues in communication and awareness.

**Basic Movements:** Due to the perspective-based use of human-scale avatars, the interactive inputs and control schemes are limited. While navigation using a mouse is possible, it may cause difficulties due to perspective issues, so it is recommended to use keyboard control schemes similar to those used in most games and visualization software. Speed adjustments like running, jumping, and ghost-flying modes are considered necessary. The ghost-flying mode allows bypassing collision calculations, passing through walls, and perceiving the project from perspectives outside the walking plane.

**Collision:** For navigating the virtual environment similarly to the physical one, collision functionality is essential. In game engines, this is achieved through "collider" calculations. This feature allows an avatar to interact with the virtual ground and prevents passing through other objects. Collision calculations are anticipated to play a crucial role in design ergonomics, spatial dimensions, and circulation issues.

**Viewing:** To enhance communication and increase awareness among users, the direction users are looking at must be reflected in their avatars' head and body movements. This can be achieved using a technique known as Inverse Kinematics (IK). This technique can also be used to animate actions like pointing with fingers.

**Stepping:** Similar to viewing, making the foot and associated body movements of the humanoid character more realistic based on the surface it is on can be achieved through procedural animation. Like viewing animations, stepping animations can also be carried out using the Inverse Kinematics (IK) method.

**Additional Animations:** To allow users to express their reactions, emotions, and thoughts, and to increase communication effectiveness in the environment, additional animations can be loaded onto avatars. Animations such as waving, clapping, nodding, sitting, and laughing can add an extra layer to communication. Such animations are common in multi-user virtual environments, and ready-made animations can be applied to a humanoid character with a rig.

**Size Manipulation:** In game engines, as in computer-aided design applications, it is possible to change the size of most objects, including avatars. To perceive the environment in different scales, avatars' sizes should be adjustable. This allows the models in the environment to be perceived as scale models, similar to those in a real studio environment, based on the avatar's current size. When avatars grow, the terrain will perceptually shrink and remain as a model, while when avatars shrink, it will be possible to perceive the fine details of detailed models. Manipulating the size of the avatar is important; while changing the sizes of objects in the environment is also possible, this change affects everyone in the studio. However, changing the size of avatars while keeping the sizes of objects fixed allows all users to view the objects at their desired scale.

## Camera

**Perspective:** To perceive the environment relative to the human scale, both third-person and first-person perspectives are required. The third-person perspective allows users to see the environment from an external viewpoint, while the first-person perspective provides the view from the avatar's eyes.

**Zoom:** Independent of perspective mode, adjustable zoom-in and zoom-out features are necessary. In third-person perspective mode, zoom allows users to view the environment from a broader perspective, while in first-person mode, it enables a closer view. Two methods for implementing zoom-in-camera programming have been identified: changing the field of view (FOV) for zoom illusions, and directly adjusting the distance between the camera and the user avatar.

**Camera Collision:** Similar to calculating collisions between objects, the necessity of calculating the camera's collision with objects and adjusting its position, especially in a third-person perspective, has been anticipated. For example, when the camera moves below the ground, it should collide with the ground plane to maintain its position above it, preventing issues like seeing the environment from below the ground or the avatar being hidden behind walls.

**Parallel Projections:** The need for parallel projection, commonly used in architectural representation, has been anticipated. However, moving in parallel projection mode can be challenging when using avatar controls and a WASD control scheme. This is due to the intersection plane created by the projection plane and the environment, and orientation issues caused by the chosen avatar and camera control scheme. It is suggested that parallel projection mode should be implemented through an additional interaction method independent of the avatar control scheme.

**Observer Mode:** To address potential orientation issues for users unfamiliar with the control schemes, a control and camera mode including orbital rotation and panning (common in computer-aided design) has been anticipated. Additionally, an observer mode, often used in games, is required to allow limited viewing modes from perspectives not possible with human perspective or from behind the avatar. In observer mode, users can follow the camera of another selected user.

**Visualization Modes:** One of the key reasons for using game engines in architecture is their real-time visualization capabilities. Game engines also offer customizable visualization through various shader programming techniques, which can be beneficial for architectural visualization. Examples of visualization modes include edge detection and highlighting, intersection detection and section creation, collision surface detection and transparency, and x-ray algorithms.

## Analysis

**Literature Review:** Various analysis methods that can be performed with the help of game engines have been identified through the examination of existing game and virtual environment software. Some of these methods are summarized below:

**Ramp Angle:** One output of collision calculations is the detection of the collided object, which allows for determining the surface normal of the accessible object. By calculating the collision at the point where the avatar steps, the angle between the surface normal and the avatar's vertical axis can be computed, allowing for the calculation of the ramp angle of the surface.

**Simple Sun Movement:** In game engines, the angle and position of an object designated as the primary light source can be adjusted. In Unity, a light source can be set as the sun, with its position determined by adjusting angles in two axes. Shadows of objects are automatically calculated as the sun’s angle changes, thanks to real-time rendering.

**Sun Path:** In addition to simple sun movement, sun path calculation algorithms can be used to determine the sun's position and trajectory based on real-world solar positions and orbits.

**Illumination and Shadows:** Game engines can calculate the number of light rays hitting surfaces from any light source. This method allows for analyzing how much light internal illuminations deliver to different points in the space and calculating annual shadow and illumination analyses for interior and exterior surfaces based on the sun's path.

**Climate:** Various climate simulations can be performed using visualization programming techniques and particle effects commonly used in game engines. Methods include simulating rain, snowfall, snow accumulation, and the effects of wind on different objects.

**Compass:** By using vectors indicating the direction the avatar is facing, it is possible to determine the user's orientation relative to the scene's north. This is useful for correcting models that may not have been aligned with the cardinal directions during the modeling phase.

**Mini-Map:** The necessity of mini-maps, which are commonly used in games for user orientation, has been identified. This allows users to check their location within the environment. Features that should be included in the mini-map are the locations of other users, the layout of the environment, and a compass-based marker or rotation of the entire mini-map according to compass directions.

**Spatial Layout:** The connections between spaces within the environment can be controlled and calculated using modules like Unity's NavMesh, which is used for calculating non-player character (NPC) movements. This calculation allows for the listing of volume connections and access to space syntax metrics.

**Pathfinding Agents:** Similar to spatial layout calculations, pathfinding algorithms and rule-based randomness programming can simulate the movement of non-player characters. This includes not only navigation but also emergency escape simulations for agents trying to reach the nearest exit from their current positions.

**Walking Distance:** Using pathfinding algorithms (such as A* and Dijkstra's algorithm), the shortest path between two points can be calculated, along with the estimated time required to traverse this path at a given walking speed. This also includes determining the maximum distance reachable within a specific timeframe.

**User Movements:** Tracking vector movements and camera orientations of users can provide insights into user circulation and the most frequently viewed areas, enabling analyses of user behavior and interaction within the environment.

**Wheelchair:** By analyzing ramp angles, constraints can be added to the character control code for wheelchair users. This includes limiting step heights and adjusting movement on steep ramps, either slowing it down or completely preventing it.

## Interactions

**Marking:** Beyond avatars' gaze, it is anticipated that marking tools are necessary for precise and accurate communication. Methods such as creating a marking object at a point identified through raycasting or visualizing the ray itself can be used. Calculations such as measuring the distance between two marks, measuring the angle between two lines using three marks, measuring elevation, and measuring surface inclination can also be performed using the same method.

**Object Library:** In visualization software, object libraries commonly consist of pre-modeled objects embedded into the program, which can be generated at desired points during a session using marking logic.

**Perspective Recording:** Recording a specific perspective angle is possible through recording the camera's current position and rotation. This allows users in observer or avatar modes to access these recorded perspectives.

**Light Object:** Another advantage of real-time visualization is the possibility of adding light objects at any moment. Users can place light objects at desired points in indoor or outdoor settings to illuminate the space. This technique also allows for night-time illumination simulations when the sun, essentially acting as a light object, sets.

**Painting:** By combining marking and collision, it is possible to change the properties of a marked surface. Surfaces can be painted, different materials can be assigned, or they can be made transparent.

**Sketching:** With the help of marking objects, three-dimensional or planar sketches can be made. Sequential marking objects can be used to draw lines and create basic geometric shapes. Additionally, perspective recording allows for freezing camera controls at a specific perspective to make two-dimensional sketches from that perspective.

### Communication

**User Names:** To enhance awareness among users and facilitate healthy communication, it is anticipated that avatars should display the names of their users.

**Voice Communication:** One of the most common methods for facilitating communication in virtual environments is voice communication. It is anticipated that voice communication, filtered based on avatar distances, can enable multiple conversations within the same virtual environment.

**Text Communication:** In situations where voice communication is unclear or persistent communication is desired, it is anticipated that text communication will be necessary.

## I/O

One of the most important features required in a virtual environment developed for architecture is digital model transfer.

**Importing:** This feature is commonly found in visualization software but is rarely present in games, and game engines do not inherently support runtime import capabilities. This feature needs to be specifically added by the virtual environment developer. After implementing import programming, the first issue encountered is file incompatibility. Game engines work only with “mesh” models. Mesh models have front and back faces, and the most important aspect is ensuring these faces are correctly configured by the user. High-polygon models (high-poly) will decrease application performance in the virtual environment, while low-detail, low-polygon (low-poly) models may not accurately represent the modeled geometry. An additional concern in multi-user environments is file sizes. To ensure that all users can see each other's models, the server must be able to deliver these files to all users. To overcome such issues and make the user experience as seamless as possible, it is anticipated that plugins capable of transferring CAD models to virtual environments in the most suitable format will need to be developed.

**Exporting:** For the analysis, simulation, interaction, or communication results produced in the environment to be transferred to other environments, the virtual environment must first support exporting. To ensure that the data exported from the virtual environment is compatible with CAD software, the data must be read and converted to one of the local data types of the drawing software.

The research is ongoing with usage tests being conducted in software development and architectural design studios. No results have yet been obtained from these tests. It is anticipated that in the future, such virtual architectural design studios will become more widespread and directly contribute to both practical and research areas of architectural education.

## Acknowledgments

The research that forms the subject of this paper is supported by the Istanbul Bilgi University Scientific Research Project (no: 2021.01.004).

## References

Bartosh, A., & Philip, A. (2019). Experimental applications of virtual reality in design education. K. Bieg, D. Briscoe & C. Odom (eds.), *ACADIA 38: Ubiquity and Autonomy* (pp. 458-467). ACADIA. http://papers.cumincad.org/data/works/att/acadia19_458.pdf

Black, C., & Forwood, E. (2017). Game engine computation for serious engineering: Visualisation and analysis of building facade movements as a consequence of loads on the primary structure. T. Nagakura, S. Tibbits & C. Mueller (eds.), *ACADIA 36: Disciplines & Disruption* (pp. 146-153). ACADIA. http://papers.cumincad.org/cgi-bin/works/paper/acadia17_146

Chien, S.-F., Lin, C.-I., & Lin, K.-L. (2020). A Light-weight CAAD-VR Bridge. D. Holzer, W. Nakapan, A. Globa & I. Koh (eds.), *CAADRIA 25: Anthropocene, Design in the Age of Humans* (pp. 761-770). CAADRIA. http://papers.cumincad.org/cgi-bin/works/paper/caadria2020_387

Coşkun, E. & Çağdaş, G. (2018). Considering computer games as a learning tool in basic design education. *International Journal Of Advanced Research*, 6(7), 1077-1095.

Du, J., Shi, Y., Mei, C., Quarles, J., & Yan, W. (2016). Communication by interaction: A multiplayer VR environment for building walkthroughs. J.L. Perdomo-Rivera, A. Gonzáles-Quevedo, C.L. del Puerto, F. Maldonado-Fortunet & O.I. Molina-Bas (eds.)

*Construction Research Congress 2016* (pp. 2281–2290). ASCE. https://doi.org/10.1061/9780784479827.227

Grasser, A., Parger, A., & Hirschberg, U. (2020). Pervasive Collaboration and Tangible Complexity in Realtime Architecture. W Liss, D. Koering (eds.), *eCAADe 2020: Anthropologic – Architecture and Fabrication in the Cognitive Age* (pp. 393-400). eCAADe. 10.14279/depositonce-10568.2

Gül, L. F. (2019). Dijital teknolojilerin işbirlikli tasarıma olan etkisi. *Megaron: Yıldız Technical University, Faculty of Architecture E-Journal 15(1),* 84-98. https://doi.org/10.14744/megaron.2019.23281

Gül, L. F. (2020). Sanal mecralarda mimari tasarımın dünü ve bugünü. *Dosya: Gelecek, Teknoloji ve Mimarlık, 45,* 28–38.

Hong, S. W., El Antably, A., & Kalay, Y. E. (2019). Architectural design creativity in multi-user virtual environment: A comparative analysis between remote collaboration media. *Environment and Planning B: Urban Analytics and City Science*, *46*(5), 826–844. https://doi.org/10.1177/2399808317733267

Leitão, A., Castelo-Branco, R., & Santos, G. (2019). Game of renders: The use of game engines for architectural visualization. M. Haeusler, M.A. Schnabel & T. Fukuda (eds.), *CAADRIA 24: Intelligent & Informed* (pp. 655-664). CAADRIA. http://papers.cumincad.org/data/works/att/caadria2019_194.pdf

Moleta, T. (2016). Game on: Exploring constructive design behaviors through the use of real-time virtual engines in architectural education. *International Journal of Architectural Computing*, *14*(3), 212–218. https://doi.org/10.1177/1478077116663341

Moleta, T. J. (2017). Digital ephemera: Autonomous real-time events in virtual environments. P. Janssen, P. Loh, A. Raonic & M.A. Schnabel (eds.), *CAADRIA 22: Protocols, Flows, and Glitches* (pp. 13-22). CAADRIA. http://papers.cumincad.org/cgi-bin/works/paper/caadria2017_158

Nandavar, A., Petzold, F., Nassif, J., & Schubert, G. (2018). Interactive virtual reality tool for BIM based on IFC - Development of openBIM and game engine based layout planning tool - A novel concept to integrate BIM and VR with bi-directional data exchange. T. Fukuda, W. Huang, P. Janssen, K. Crolla & S. Alhadidi (eds.), *CAADRIA 23: Learning, Prototyping and Adapting* (pp. 453-462). CAADRIA. http://papers.cumincad.org/cgi-bin/works/paper/caadria2018_057

Pienaru, M.-I. (2018). The city as a playground - game tools for interactive planning. A. Kepczynska-Walczak & S. Bialkowski (eds.), *eCAADe 36: Computing for a Better Tomorrow* (pp. 679-686). eCAADe. http://papers.cumincad.org/cgi-bin/works/paper/ecaade2018_375

Sandstrom, A., & Park, H.-J. (2019). Reflection in action - An educational indie video game with design schema. M. Haeusler, M.A. Schnabel & T. Fukuda, (eds.), *CAADRIA 24: Intelligent & Informed* (pp. 303-312). CAADRIA. http://papers.cumincad.org/cgi-bin/works/paper/caadria2019_176

Sorguç, A. G., Yemişcioğlu, M. K., Özgenel, Ç. F., Katipoğlu, M. O., & Rasulzade, R. (2017). The role of VR as a new game changer in computational design education. A. Fioravanti, S. Cursi, S. Elahmar, S. Gargaro, G. Loffreda, G. Novembri, A. Trento (eds.), *eCAADe 35: ShoCK! - Sharing Computational Knowledge!* (pp. 401-408). eCAADe. http://papers.cumincad.org/cgi-bin/works/paper/ecaade2017_142

Valls, F., Redondo, E., Fonseca, D., Garcia-Almirall, P., & Subirós, J. (2016). Videogame technology in architecture education. *Human-Computer Interaction: Novel User Experiences,* 9733, 436–447. https://doi.org/10.1007/978-3-319-39513-5_41

Yan, W., & Liu, G. (2007). BIMGame: Integrating building information modeling and games to enhance sustainable design and education. J. Kieferle & K. Ehlers (eds.), eCAADe 25: Architecture in the Age of the 4th Industrial Revolution (pp. 211-218), eCAADe. http://papers.cumincad.org/cgi-bin/works/paper/ecaade2007_188