ISTANBUL BILGI UNIVERSITY
INSTITUTE OF GRADUATE STUDIES
HISTORY, THEORY AND CRITICISM IN ARCHITECTURE
MASTER'S DEGREE PROGRAM

MULTI-USER VIRTUAL ENVIRONMENTS IN ARCHITECTURE

Oğulcan Üneşi
118803012

Prof. Dr. A. Tuğrul Yazar

ISTANBUL
2022

MULTI-USER VIRTUAL ENVIRONMENTS IN ARCHITECTURE

MİMARLIKTA ÇOKLU-KULLANICILI SANAL ORTAMLAR

Oğulcan Üneşi

118803012

Thesis Supervisor:	Prof. Dr. A. Tuğrul Yazar Istanbul Bilgi University	....
Jury Members:	Prof. Dr. Leman Figen Gül Istanbul Technical University	....
	Doç. Dr. Fulya Akipek Istanbul Bilgi University	....

Date of Approval: 26/09/22

Total Page Count: 253

Keywords (English)
1) Distance Education
2) Design Studio
3) Architectural Education
4) Game Engines
5) Game Technologies

Keywords (Turkish)
1) Açık ve Uzaktan Öğrenme
2) Tasarım Stüdyosu
3) Mimarlık Eğitimi
4) Oyun Motorları
5) Oyun Teknolojileri

ACKNOWLEDGEMENTS

First of all, I would like to express my sincerest gratitude to my thesis supervisor, Prof. Dr. Tuğrul Yazar, whose immeasurable guidance, encouragement and support has made this research possible.

I would like to express a special thank you to the jury members, Prof. Dr. Leman Figen Gül and Doç. Dr. Fulya Akipek for their invaluable comments and guidance, and their approval for the extension period that was necessary to conclude this research.

I would like to thank Istanbul Bilgi University for supporting this research through another study made in parallel, as a part of Scientific Research Projects (BAP).

I would also like to thank all of the participants of this research, without whom I would not have been able to complete this research.

I am extremely grateful to my friends, Berk Meydanlı, Halil İbrahim Kaya and İbrahim Serkan Uysal for their contributions in testing and debugging for the software, and their immense support throughout this research. I would also like to extend my thanks to Aşkın Aydın for his support and mentorship in my times of need.

From the bottom of my heart, I would like to thank my parents, Aysun Tezgeç and İsmail Cem Üneşi, for their unconditional love, and support of my pursuits.

Finally, I would like to express my gratitude and appreciation for my partner, Kübra Koyuncu, for her endless love and support, and being in my life.

Oğulcan Üneşi
November 2022

ABSTRACT

MULTI-USER VIRTUAL ENVIRONMENTS IN ARCHITECTURE

The necessity of remote working and distance education tools have become undeniable during the course of the COVID-19 pandemic. The sudden requirement of appropriate methods and tools have caught architectural design studios by surprise. This has caused some difficulties in adapting the traditional architectural design studios to remote working and distance education concepts, which brought the implication of the need for specialized tools for such scenarios in certain disciplines.

This research examines the difficulties faced in architectural design studios during the distance education period and the literature on the use of virtual environments and remote working technologies in architecture and reveals the development possibilities. As a result of the study, the perceptual and technological problems, deficiencies and potentials of this field were emphasized and discussed. It is predicted that in the future, there will be more need for remote working and distance education technologies, through virtual environments for various reasons. This study aims to serve as a guide for the development of new virtual environment technologies in the future.

Keywords: Distance Education, Design Studio, Architectural Education, Game Engines, Game Technologies

ÖZET

MİMARLIKTA
ÇOKLU-KULLANICILI
SANAL ORTAMLAR

COVID-19 pandemisi sürecinde uzaktan çalışma ve uzaktan eğitim araçlarının gerekliliği yadsınamaz hale gelmiştir. Aniden uygun yöntem ve araçlara duyulan ihtiyaç, mimari tasarım stüdyolarını hazırlıksız yakalamıştır. Bu durum, geleneksel mimari tasarım stüdyolarının uzaktan çalışma ve uzaktan eğitim kavramlarına uyarlanmasında bazı zorluklara neden olmuş ve de belirli disiplinlerde bu tür senaryolar için özel araçlara ihtiyaç duyulduğunu ortaya çıkarmıştır.

Bu araştırma, uzaktan eğitim döneminde mimari tasarım stüdyolarında karşılaşılan zorlukları ve sanal ortamlar ile uzaktan çalışma teknolojilerinin mimarlıkta kullanımına ilişkin literatürü incelemekte ve bu alandaki geliştirme olanaklarını ortaya koymaktadır. Çalışma sonucunda bu alanın algısal ve teknolojik sorunları, eksiklikleri ve potansiyelleri üzerinde durulmuştur. Gelecekte çeşitli nedenlerle sanal ortamlar aracılığıyla uzaktan çalışma ve uzaktan eğitim teknolojilerine daha fazla ihtiyaç duyulacağı öngörülmektedir. Bu çalışma, gelecekte yeni sanal ortam teknolojilerinin geliştirilmesine yönelik bir rehber niteliği taşımayı amaçlamaktadır.

Anahtar Kelimeler: Açık ve Uzaktan Öğrenme, Tasarım Stüdyosu, Mimarlık Eğitimi, Oyun Motorları, Oyun Teknolojileri

TABLE OF CONTENTS

ACKNOWLEDGEMENTS **iii**

ABSTRACT **iv**

ÖZET **v**

TABLE OF CONTENTS **vi**

LIST OF FIGURES **viii**

LIST OF TABLES **xii**

LIST OF SYMBOLS / ABBREVIATIONS **xiii**

1. INTRODUCTION **1**

1.1. The Tools and the Boundaries 1

1.2. The Digital Realm 2

1.3. Information Processing 3

1.4. Post-Digital 4

1.5. Remote Working 5

1.6. The Aim of the Research 8

1.7. Objectives 9

1.8. Research Questions 10

1.9. Hypotheses 11

1.10. Methodology 12

1.11. The Process 15

2. LITERATURE REVIEW **17**

3. PRELIMINARY STUDIES **31**

3.1. Student Survey 31

3.2. Design Studio Tutor Interview 69

3.3. Student Interview 74

3.4. Findings 83

3.5. State-of-the-Art 93

4. FEATURE LIST **114**

4.1. Filtering and Constraints 118

4.2. Expansion 119

4.3. Technical Design 130

4.4. The Feature List 136

5. DIGITAL DESIGN STUDIO CASE STUDY **146**
5.1. Second Design Studio Tutor Interview 147
5.2. Third Design Studio Tutor Interview 156
5.3. Designing the Presentation Environment 164
5.4. Test Environment 164
5.5. OVS 165
5.6. Studio Sessions 183
5.7. Feedback Survey 190
5.8. Online Virtual Jury 208

6. CONCLUSIONS **223**
6.1. Findings 225
6.2. Next Steps 228
6.3. Regarding OVS 229
6.4. Discussion 230

REFERENCES **232**

URL REFERENCES **236**

LIST OF FIGURES

Figure 3.1. Answers for Question 1 33
Figure 3.2. Answers for Question 2 34
Figure 3.3. Answers for Question 3 36
Figure 3.4. Answers for Question 4 37
Figure 3.5. Answers for Question 5 38
Figure 3.6. Answers for Question 6 41
Figure 3.7. Answers for Question 7 42
Figure 3.8. Answers for Question 8 43
Figure 3.9. Answers for Question 9 44
Figure 3.10. Answers for Question 10 45
Figure 3.11. Answers for Question 11 46
Figure 3.12. Answers for Question 12 47
Figure 3.13. Answers for Question 13 48
Figure 3.14. Answers for Question 14 49
Figure 3.15. Answers for Question 15 50
Figure 3.16. Answers for Question 16 51
Figure 3.17. Answers for Question 17 52
Figure 3.18. Answers for Question 18 53
Figure 3.19. Answers for Question 19 54
Figure 3.20. Answers for Question 20 56
Figure 3.21. Answers for Question 21 57
Figure 3.22. Answers for Question 22 58
Figure 3.23. Answers for Question 23 59
Figure 3.24. Answers for Question 25 61
Figure 3.25. Answers for Question 27 63
Figure 3.26. Answers for Question 28 64
Figure 3.27. Pointers of Other Users in Miro 94
Figure 3.28. Wardrobe in Cyberpunk 2077 95
Figure 3.29. Players Dancing in a Deep Rock Galactic Lobby 96
Figure 3.30. Foot IK in Grand Theft Auto Online 96
Figure 3.31. Pointing in Grand Theft Auto Online 97
Figure 3.32. TPP-FPP switch in Grand Theft Auto V 98
Figure 3.33. PC vs VR in Davigo 99
Figure 3.34. Spatial Running on an Internet Browser 100
Figure 3.35. Voice Chat Bindings in Phasmophobia 101
Figure 3.36. (a) Grenade Trajectory in Cyberpunk 2077 (b) Invalid Placement of a Charcoal Kiln in Valheim 102
Figure 3.37. Various Stages of a Multiplayer Game in Risk of Rain 2 104

Figure 3.38. Importing 2D and 3D Objects in Spatial 105
Figure 3.39. Building Prefabs in Valheim 106
Figure 3.40. Parametric Objects in Arkio 107
Figure 3.41. Participants Annotating in Zoom 107
Figure 3.42. Laser Pointers in Deep Rock Galactic 108
Figure 3.43. (a) 3D Map and Routes in Cyberpunk 2077
(b) Minimap in Cyberpunk 2077 109
Figure 3.44. OpenStreetMap Import With Buildings in Arkio 110
Figure 3.45. Various Info-Views in Cities: Skyline 111
Figure 3.46. A Street Crowded with NPCs in Cyberpunk 2077 111
Figure 3.47. Day and Night in Night City, in Cyberpunk 2077 112
Figure 3.48. Cross Section Shaders by Abdullah Aldandarawy 113
Figure 3.49. Snow in Monster Hunter: World 113
Figure 5.1. Humanoid Avatar 147
Figure 5.2. Flying Avatar 148
Figure 5.3. Foot IK 148
Figure 5.4. Super-Sized Avatar 149
Figure 5.5. Micro-Sized Avatar 149
Figure 5.6. (a) First-Person and (b) Third-Person Perspectives of the same view 150
Figure 5.7. Pointing and spawning a temporary tag 151
Figure 5.8. A model with several annotations 151
Figure 5.9. Controlling the Sun 152
Figure 5.10. Wheelchair Simulation 153
Figure 5.11. Rainwater Accumulation 154
Figure 5.12. OVS Demonstration Video Conference 154
Figure 5.13. A Multi-User Session 157
Figure 5.14. In-session Pause Menu 158
Figure 5.15. Taking Notes Inside a Permanent Marker 158
Figure 5.16. Pop-Up Note On Top of a Permanent Marker 159
Figure 5.17. A Display Board With a 2D Asset 160
Figure 5.18. A 3D Model of the LC-4 Lounge Chair in the Environment 161
Figure 5.19. The Presentation Setup 162
Figure 5.20. WIP Airsketch Tool 163
Figure 5.21. OVS Distribution 166
Figure 5.22. Landing Menu 166
Figure 5.23. Blank Environment 167
Figure 5.24. Connection Menu 168
Figure 5.25. Import/Export Tab in In-Session Context Menu 169
Figure 5.26. Loading Model in Progress 169
Figure 5.27. Imported Model and Environment 170

Figure 5.28. Help Overlay 171
Figure 5.29. In-Session Menu 171
Figure 5.30. Feedback Survey 172
Figure 5.31. Context Menu 172
Figure 5.32. Analysis Tab 173
Figure 5.33. User Trail 174
Figure 5.34. Simulation Tab 174
Figure 5.35. An Avatar on a Wheelchair with no HUD 175
Figure 5.36. Various Avatars with Name Tags 175
Figure 5.37. Waving Animation 176
Figure 5.38. User Avatar in 1:1 Scale 176
Figure 5.39. User Avatar in 20:1 Scale 177
Figure 5.40. First Person Perspective 178
Figure 5.41. Parallel Projection Camera on the Posters 178
Figure 5.42. Pointing and Placing a Temporary Marker 179
Figure 5.43. Toolbar (a) Temporary Marker Selected
(b) Permanent Marker Selected (c) Elevation Marker Selected 180
Figure 5.44. Typing a Note Inside a Permanent Marker 180
Figure 5.45. Pop-Up Text on a Permanent Marker 181
Figure 5.46. Permanent Marker Delete Confirmation 181
Figure 5.47. 3D Polylines, Distance Measurement and Elevation Markers 182
Figure 5.48. Airsketch Tool 182
Figure 5.49. Airsketch Marker 183
Figure 5.50. Introduction to OVS 184
Figure 5.51. Virtual Field Trip 184
Figure 5.52. OVS in a Face-to-Face Setting 186
Figure 5.53. Posters Inside a Project 187
Figure 5.54. Using Airsketch to Give Criticism 188
Figure 5.55. Physical Media and OVS 188
Figure 5.56. Color-Coded Paths 189
Figure 5.57. Answers for Question 2 192
Figure 5.58. Answers for Question 3 193
Figure 5.59. Answers for Question 6 195
Figure 5.60. Answers for Question 7 196
Figure 5.61. Answers for Question 8 197
Figure 5.62. Answers for Question 9 198
Figure 5.63. Answers for Question 10 199
Figure 5.64. Answers for Question 11 200
Figure 5.67. Answers for Question 16 205
Figure 5.68. The Presentation Environment 209

Figure 5.69. A Model Display 210
Figure 5.70. (a) Tutors Looking at the Brief
(b) Viewing the Poster in Parallel Projection Camera 202
Figure 5.71. Student Presenting Their Work on Posters 212
Figure 5.72. Visual Obstruction by Avatars 213
Figure 5.73. Participants Utilizing Avatars and Pointing 214
Figure 5.74. Participants Flying Inside the Model of a Project 214
Figure 5.75. Participants Meeting in the Park 215
Figure 5.76. Supersized Avatars 216
Figure 5.77. Exploring the Interiors With a Reference to Human Scale 217
Figure 5.78. Flying Supersized Avatars Clipping Through the Ground 218
Figure 5.79. Participants Moving On to the Next Project 219
Figure 5.80. A Model With Placeholder Textures 221

LIST OF TABLES

Table 4.1. “Character” category of the feature list 137
Table 4.2. “Character Animations” category of the feature list 138
Table 4.3. “Camera” category of the feature list 139
Table 4.4. “User Interface” category of the feature list 140
Table 4.5. “Analysis” category of the feature list 141
Table 4.6. “Simulation” category of the feature list 142
Table 4.7. “User Interaction” category of the feature list 143
Table 4.8. “Communication, AI and Networking” categories of the feature list 144
Table 4.9. “Input/Output and Scene” categories of the feature list 145

LIST OF SYMBOLS / ABBREVIATIONS

%	Percent
2D	2 Dimensional
3D	3 Dimensional
3DM	OpenNURBS File Format
AEC	Architecture, Engineering and Construction
AI	Artificial Intelligence
API	Application Programming Interface
APP	Application
AR	Artificial Reality
BIM	Building Information Modeling
C#	C# Programming Language
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
CAT	Consensus Assessment Technique
DWG	Autodesk .dwg File Format

FBX	Autodesk .fbx File Format
FPP	First-Person Perspective
FPS	First-Person Shooter
FTP	File Transfer Protocol
GIF	Graphics Interchange Format
HID	Human Interface Device
HUD	Heads-Up Display
IDE	Integrated Development Environment
IFC	Industry Foundation Classes File Format
JPEG	Joint Photographic Experts Group File Format
JPG	Joint Photographic Experts Group File Format
JSON	Javascript Object Notation File Format
MMO	Massively Multiplayer Online
MR	Mixed Reality
NPC	Non-Player Character
NURBS	Non-Uniform Rational B-Spline
OBJ	Wavefront .obj File Format

P2P	Peer-to-Peer
PNG	Portable Network Graphics File Format
PREFAB	Prefabricated (Game) Object
PVP	Player-Versus-Player
QoL	Quality of Life
RPG	Role-Playing Game
RTS	Real-Time Strategy
SDLC	Software Development Life Cycle
TPP	Third-Person Perspective
TPS	Third-Person Shooter
UI	User Interface
UX	User Experience
VR	Virtual Reality
WIP	Work-in-Progress
XR	Extended Reality

1. INTRODUCTION

Architecture has always been mutually related to technology. Whether they are in construction technologies, material research or developments, or in the way we design, technological advances have been influencing the course of architecture throughout history. Although it may seem that technology guides architecture, pioneering architects have been influencing or directly orchestrating technological advancements through their vision and designs. Similar to how Brunelleschi had to solve the complex problem of building his dome by inventing a novel construction system, architects today still face situations where the need for a new type of material or a system has to be developed in order to realize the idea. Even though the description and the scope of the architect *now* are not what they used to be *then*, such situations require the mindset of the *uomo universale.*

1.1. The Tools and the Boundaries

Regarding the pace of the technological advancements of our age, it wouldn't be right to assume that it is progressive to grow comfortable in doing what we know best and what's available in our architectural comfort zones, both in education and practice. Following the most recent developments and adapting where it is comfortable is hardly the same as building the dome. That being said, it would also be hard to deny the importance of boundaries in architectural design; context, requirements, and the tools we are given define what can be done. Designing with endless possibilities on an endless plane would be fruitless if not impossible. We do need to have ties with reality but that doesn't mean we have to restrict ourselves only to what's available. Being too ambitious can receive a negative response and the solid distinction we see between engineers and architects today can be disheartening, although that distinction might not be as strict as some might perceive.

Architects build by designing; simply put, by drawing. As we proceed through time, our drawing tools have become more and more complex to the point where we don't even have the slightest idea of how they work. They now require expertise in separate engineering fields which depend on each other to ultimately shape our tools and keep them working. Hardware is designed and built, operating systems make the hardware have basic functionality and software gives us the drawing tools. All of these are unknown waters for most architects, even without mentioning human-computer interface devices such as the mouse, the keyboard, the screen, and the basic infrastructure for all of this to work such as electricity and optionally the internet, trying to absorb all of the technical details would be overwhelming. An individual architect cannot possibly achieve basic knowledge in all of these fields, although that shouldn't be necessary either. However it is possible to gain knowledge in the last step before becoming the *end-user,* software development, which is getting more and more common in almost every field. Through the knowledge gained in software development, a group of individuals should be able to develop the necessary tools specifically tailored for their own use.

1.2. The Digital Realm

Aside from the advancements in construction techniques and materials which help actualize the design and shape the boundaries, the Information Age has opened up a new domain for architectural design. The seeds sown by Ivan Sutherland's "Sketchpad" have grown and branched into a vast field employed by many disciplines (Sutherland, 1964). Since personal computers entered our homes, affordable or free mainstream computer-aided design (CAD) software have made it into our personal computers. The same ambition of Gaudí with his hanging chain models has attracted the curiosity of architects to using such software (Kandela, 2001). Organic forms which were difficult to express in conventional representation techniques have found their way into architecture, tough and repetitive tasks were suddenly easy and a few chains that held back the architect were broken as a result. Apart from the changes in the representation techniques, this development also affected the theory and culture of architecture. In his dissertation, Joao Rocha reveals the historical background of the computational perspective architecture has gained (Rocha, 2004).

Today, using CAD software has become the norm in architectural offices and universities and proficiency in popular drawing software is commonly expected in the field.

Computer-aided drawing and modeling is just the bare minimum of what can be done today, with recent breakthroughs in manufacturing methods and technologies, computer-aided manufacturing (CAM) has also made it into our homes with household appliances such as affordable 3D printers. Small-scale 3D printers aside, large-scale 3D printers, uses of drones and industrial robots have advanced the use of CAM in architecture exceptionally.

1.3. Information Processing

The transition from physical to digital may only seem like a shift of the medium at first glance. In addition to the sharp and accurate lines, faster operations, and brand-new representation techniques, the infinite universe of computation and information processing became available to architects as the ultimate result. Once the design was formed in the digital realm, everything else in this realm could interact with it. The various analyses could be performed digitally, time-consuming calculations could be done in seconds, analysis and simulation software developed for technically more advanced fields became available to all architects. In the study titled: “Theory and Design in the First Digital Age”, Oxman identifies the key elements of digital design processes and their combinations (Oxman, 2006). Today, most of the arguments in this study have been proven to be correct.

Recently, such analysis and simulation software have also gained widespread use and official recognition in architecture. 3D modeling in combination with such methods have evolved into building information modeling (BIM), an almost all-in-one approach to plan, design, construct and manage. Computational design, which became accessible with the digital transition, has also taken a peculiar turn. Heavily inspired by computer sciences, the parametric design emerged from design algorithms that consist of sequential parametric equations and operations. Architects using such methods have had their role shifted from designing the actual project to designing the formula/script which would “solve” the equation when executed. Aish explains this with the shift of focus in architecture, from singular objects to processes (Aish, 2005). Furthermore, it isn’t considered uncommon to tie these scripts to the aforementioned analysis and simulation methods in an attempt to “optimize” the design (or the equation) in a data-informed manner. Utilizing computational design on this level and enabling programming scripts in architectural design brought the field of architecture inside of the scope of automation and artificial intelligence.

Today, in the trending field of artificial intelligence and more importantly, machine learning, there is promising research being conducted in the field of architecture (Chaillou, 2019). Even though the number of studies is rapidly increasing on this topic, the application of such technologies in practice is not common yet. It is possible to suggest that this would be one of the core topics in the future of architecture, though it shouldn't be controversial to suggest more emphasis on the subject in basic architectural education besides the postgraduate level. However, it is important to note that attempting to do so would require the preceding steps to be taken first.

1.4. Post-Digital

It is predicted that the tools architects are using today will influence the course of architecture into an even more computation-oriented, AI driven direction due to the amount and the nature of the processing power that is at our disposal (Carpo, 2017). This also points to the paradigm shift in the role of the architect. Instead of putting effort into solving the actual architectural problems through design, the architect would have to assume the role of the *programmer* by defining the equation or the *script* and use the computer to find the solution or a variety of solutions to choose from. This has already been the case in the use of the most recent technologies in architecture, such as *machine learning* and *generative design*.

Along with the progression towards this prediction, another attitude has presented itself through digital platforms such as KooZA/rch (URL-1) albeit being limited to the form of representation. The "Post-Digital Drawing" movement influences architects to imitate the use of formerly popular techniques such as collage and hand drawing through digital tools. Approaching architectural representation in a more artistic way, "Post-Digital" renderings tend to feature narratives and appear more similar to paintings than technical drawings. While this approach may seem like only an artistic choice at first, it provides a strong alternative to realistic rendering in less refined digital drawings that lack the detail for such realism, which is more common in small offices and student projects. This brings the question of whether state-of-the-art technologies always suit the purpose. Using a hyper-realistic rendering engine on a conceptual 3D model or using AI to solve a simple problem is equally inappropriate.

In addition, even though this trend does not question the course of architecture directly, it does show segregation in the field. Where some architects have been focusing on computational design and programming, others are more concerned with artistic expression through architectural representation techniques.

1.5. Remote Working

Due to the recent COVID outbreak of 2019, companies have switched to full or scheduled remote working environments; meetings, conferences and symposiums have migrated to online platforms and education had its first term of online classrooms. Even though it is a tough time to live in, migrating to the online world in work and education has brought some advantages as well as disadvantages. It wasn't common to have a foreign professor connect to your conference via Skype, schedules had to be adjusted, plane tickets had to be bought and accommodation had to be arranged. Morning rituals had to be commenced and traffic had to be tackled to get to work or school. In such a short period of time, it became a part of our daily lives to work or listen to such lectures from the comfort of our homes. While the circumstances had us locked-in and apart from each other, we were also more accessible than ever.

This, now common, method of working has actually been possible for a while. The digital realm has been conquering our lives the internet has been connecting us in this realm. This type of communication when boosted with cloud services gives us instant access to our data, to our work in this case, which makes it possible to work from anywhere and see the same results as anyone we choose to, almost instantly. There is one factor worth mentioning at this point, the difference between a (*real-time*) collaborative environment and a synchronized cloud.

While cloud systems have become a part of our daily lives, collaborative environments are not as common. In layman's terms, a cloud system can synchronize your data and keep a collective database for multiple users, though two users modifying the same data at the same time will conflict and the actions will not be synchronous during the access phase. In such situations, more common than not, the data will be overridden with the last modification. This means if two people are working on the same document in a non-collaborative environment, such as storing a poster image in Google Drive (URL-2), starting from the moment both people access the document, not only these two people won't see the changes each other make, the last person to save the document will override the changes made by the previous person. On the other hand, if these two people were working in a collaborative environment, such as Google Docs (URL-3), even the momentary changes would instantly be visible by both parties and no data would be lost. This also provides the opportunity to efficiently communicate in such situations, imagine writing a little note under where the other person is typing or directly rejecting their editing and deleting their text as they type, although commenting or text messaging is usually present in such software. The opportunities provided by real-time collaborative environments will additionally be discussed further.

Remote working in the information era has many benefits as hinted above, yet these aren't the only ones worth mentioning. Similar to how we can listen to a foreign professor give a lecture on the opposite side of the world in real-time and ask them questions when possible, this remarkable possibility could be taken further to provide universal access from any point in the world to education that is given in any part of the world. Many online courses do provide such sessions with lecturers aside from the obvious use of online videos, although it would be hard to imagine this process being effective in design studios.

While most of the architects are familiar with recent technologies, computers, online communication services, CAD, 3D models and rendering software; architectural design phases, both in professional and educational environments, still require the "pen and paper" physical approach to communicate efficiently. Holding a physical model in your hands and pointing with your finger, cutting or breaking a piece of the material on the spot to demonstrate a point or putting a piece of material on the model to express an approach is still an easier approach than looking and pointing on a computer-generated 3D model, in terms of efficient communication. Printing the computer-generated plans and sections and marking them with a pen is still easier than grabbing the mouse and trying to draw on the screen before losing the idea. Perhaps the most dramatic discrepancy is that although we make the most detailed models of our projects, we limit them to a few rendered frames or a few short clips to be able to present them, even though the technology to see and experience it in the first-person point of view exists.

This points to the actual problem of how a conventional remote working environment isn’t sufficient for architectural design. While communicating in the physical world, we can manipulate the objects around us, use hand gestures and have a three-dimensional space to express an idea, whereas, in a conventional remote working environment, we only have our two-dimensional representation materials to aid us to communicate. Of course, we can plug in a camera, have a live conversation while going over the presentation, annotate the presentation through various online communication software and even send the original 3D model for the other side to view. Although this still limits the real-time (synchronous) communication through representation to 2D; and the 3D, more interactive representation to asynchronous.

As it is previously mentioned, technology to experience 3D models is present. In addition to those, real-time rendering engines and furthermore, *interactive* real-time rendering engines with *multi-user, collaborative* capabilities exist not too far from our daily lives, although it may not be obvious at first, these are *multiplayer videogames*. Moreover, just as almost everything in the digital realm is, they are, and perhaps most, subjective to all methods of computation in this realm.

Numerous studies have been made on the use of videogames and game engines in architecture, which will be thoroughly examined in this research. It is also worth mentioning that some architectural rendering engines such as *Lumion* (URL-4) and remote collaboration applications such as *Spatial* (URL-5) are based on (made with) game engines.

1.6. The Aim of the Research

Certain difficulties were faced during the remote working period, some of them were directly related to the medium, the environments used for communicating and collaborating, the software. Among these difficulties, there were some aspects in the methods and environments involved which could be considered as potential advantages. As there wasn't, and currently isn't, a single commonly-accepted method and medium for remote working and distance education in architecture, each method and medium have their own advantages and disadvantages. While certain media are better suited for basic communication such as verbal and written, some perform better in spatial communication but might require additional hardware such as VR environments and headsets.

From the experiences gained in the remote working period during the course of the pandemic, it was determined that a conventional remote working environment isn't sufficient for architectural design studios, both in practice and education. Keeping the aforementioned points in mind, this research takes on the assumption that this problem could be overcome through the involvement of architects in the design of such an environment.

Through the identification of the problem, this research focuses on the use of virtual environments in architectural practice and education. Studies conducted on the use of such software and environments and related technologies will be thoroughly examined. Interviews will be conducted to accurately identify the problems and the requirements of such an environment. As a result of the identification, possible solutions will be suggested and various experiments and case studies will be held to examine these methods. Advantages and disadvantages of various approaches will be determined and discussed. The aim of this research is to serve as a guideline for future software, environments and methods capable of providing a remote working and distance education environment for architects and architecture students.

1.7. Objectives

Towards achieving the aim of the research, several objectives and research interests were defined.

O1. The first of these objectives is the accurate identification and definition of the problems faced during the remote working period in architectural design studios. Attached to this first objective is the classification of these problems in terms of the cause of the problem. The problems that derive from technical difficulties will be the main focus of this objective. After the identification of these technical difficulties, possible solutions from a technical point of view will be discussed. It is assumed that at this point in research, additional potentials of virtual environments in terms of architectural design and education will be revealed.

O2. The second objective is to test a customized virtual environment software and related studio setting for the solution of these problems, which architectural design studios can benefit from in the distance education model, for the needs and methods of architectural design studios. This phase will implement a feedback loop in which the methods will be improved and re-visited upon the feedback of the testers. It was suggested that this virtual environment software should be utilizing the state-of-the-art technologies which could benefit architectural design and education, through employing a video game engine. This suggestion was later confirmed by the initial literature review and the inspection of the existing virtual environments.

O3. The third objective is to question the usability of various interaction, computation and collaborative design possibilities made possible by game engines in architectural design studios and revealing the underlying factors of this issue. This objective is detached from the "remote" aspect of the issues such as remote working and distance education and is purely focused on the potentials of such tools in architecture.

Upon defining the objectives, two axes of the research can be distinguished:

A1. Identifying the problems of remote working and distance education in architectural design studios and developing solutions, in terms of methods and tools. (In line with O1 and O2)

A2. Examining the potential contributions of tools, environments and methods to be produced and tested for this purpose to architecture, independent of remote working and distance education. (In line with O2 and O3)

1.8. Research Questions

Having described the objectives and defined the axes, three research questions emerge in correspondence.

Q1. What are the challenges faced by traditional architectural design studios in the distance education model and what are the opportunities this model offers?
(Responding to O1)

Through finding out the answers to this question, the expected features of the software that the research will test for use will be clarified.

Q2. For what purposes and to what extent can the potentials offered by virtual environments and game engine technologies be used in architecture, architectural education and distance education model?
(Responding to O2 and O3)

By answering this question, a unique guideline for developing specialized virtual environments for architecture will be defined. Expected features will be tested through the use of game engines and whether these features are useful in architectural design studios and remote working/distance learning situations will be examined.

Q3. How can a digital/virtual architectural design studio contribute to architecture and architectural education?
(Responding to O3)

The answer to this question depends on the outcome of the second question. It is predicted that a specialized virtual environment for architectural design studios will also contribute to the field of architectural education research. It is also predicted that the outcome of this question will have implications on how much the architectural design studio can "digitize" or "hybridize" as a concept.

1.9. Hypotheses

Three hypotheses were assumed for the research, the first two providing discussion grounds for the research questions and the last one for the whole process and the methodology of this research.

H1. Traditional architectural design studios have qualities that are difficult or impossible to implement with the standard toolkit of any remote working platform. Considering the visual and intellectual characteristics of the design, the qualities that representational environments must have, and the possibilities of using computational technologies, common remote working environments lack the necessary tools for supporting architectural design studios.

The first hypothesis is tied to Q1. It will be tested through the first research question. Upon finding out the answers to the first research question, Q2 will try to be answered.

H2. Considering the possibilities such as various interaction, simulation and analysis methods offered by virtual environments and game engine technologies, these technologies have qualities and opportunities that are not possible in traditional architectural design studios.

The second hypothesis is tied to Q2 and Q3. It will partially be tested through the second research question and completely by the third research question.

H3. Architects have lagged behind in developing software for themselves. There are intellectual, productive and design barriers that arise from architects being limited to the tools that they did not produce or develop. In the case of producing tools and mastering the *tools that produce tools*, it is possible to perceive possibilities that cannot be grasped otherwise.

The third hypothesis isn't tied to any of the research questions or objectives and it will be observed by the researcher through the changes in their own perception. This hypothesis will be tested through the course of the research and the chosen methodology. It will be discussed independently of the outcome of the research questions.

1.10. Methodology

In this research, a software specialized for architectural design studios being developed with a game engine, will be tested in an architectural design studio setting.

Architectural design studios predominantly rely on design dialogue between the students and the tutors. This dialogue often happens in the form of a desk review, utilizing several types of media to communicate on the design. Yazar comparatively explains the various educational approaches in architectural design studios and the fundamental aspects of design studios in their dissertation (Yazar, 2009). Architectural design depends heavily on drawings that represent the design, rather than written or verbal statements. The developing design or the design intentions have to be appropriately represented in order to be able to be communicated and discussed. The media used for representation in architectural design studios, often simply referred to as "drawings", can be in several forms such as: sketches, photographs, videos, 2D drawings done either by hand or with CAD software, 3D computer generated models, physical models or any other media that can accurately represent the design or the intentions. In conventional, face-to-face design studios, these media can easily be observed and discussed around a table, either through the physical media themselves or on a computer screen that is present in the room. Physical media can directly be marked or altered during the conversation, and digital media can either be annotated in the digital environment or the digital contents can be captured and plotted, and more recently 3D printed, to be used in the physical realm.

Game engines are software frameworks consisting of tools and features, initially designed for video game development. These frameworks often include basic video game development necessities such as input systems (as in human interface device/HID input), physics engines, 2D and 3D graphics rendering capabilities, sound input and output, animation systems, networking frameworks and most importantly, scripting possibilities.

Recently, game engines such as Unity and Unreal Engine have started to be used in various other industries such as automotive, manufacturing, film, animation, architecture, engineering and construction (URL-6). This can be tied to the fact that they offer tools and conveniences that are not only needed for video games but for any real-time software utilizing the aforementioned capabilities. Coupled with engine-specific plugins and tools or code libraries for the supported scripting languages, game engines can be used to produce almost any kind of software. As a side note, they can also be used to port the developed software into supported platforms such as Windows, Mac, iOS or Android with little or no changes in the source code except for the input handling the devices require.

One important fact to note at this point is that game engines are *only the frameworks* used to develop such software, not the software themselves. There of course are debugging tools in game engines, such as “Play Mode”, to test the developed software but in the end, the software has to be *compiled* in order to be distributed as stand-alone software. It was observed in the initial literature review that the distinction between utilizing a game engine and developing a software with a game engine is often overlooked and utilizing a game engine is sometimes confused with developing the software. It is also worth noting that some features such as networking absolutely require the software to be built and some features such as 3D model importing can be done easily in the engine itself but not in the built software without custom solutions.

In an architectural point of view, using game engines is not the only solution to provide capabilities such as multi-user interaction. Improving upon existing CAD software through plug-ins could theoretically bring such capabilities directly into a CAD environment. The acquisition of The Wild, a VR collaboration platform for the AEC Industry, by Autodesk (URL-7) could be considered a step towards the realization of such a concept. Bringing these capabilities directly into a CAD environment could eliminate the need for import/export operations and related pipeline processes. Although, the development of a software detached from any CAD software would make it possible to work with generic file formats thus providing support for many CAD software at once. This should eliminate the need for the use of a specific CAD software in order to have these additional capabilities.

Due to the testing of a software in its development cycle, a unique workflow similar to those frequently used in software development phases was proposed in order to test the hypotheses. Also known as “Software Development Life Cycle” or SDLC, these processes typically consist of various steps in linear succession or within loops in some stages of development (CMS, 2008).

There are many approaches and sets of practices regarding SDLCs and the number of steps and loops in each of these approaches vary substantially. These steps can roughly be categorized as investigation, requirements definition, feasibility analysis, design, prototyping, implementation, testing, maintenance and support. Usually, the first step of a SDLC is the initial investigation. In this phase, the problems are investigated and fully understood in order to be able to propose a solution. Responding to the first step, is the requirements definition. In this step, solutions are proposed to the problems found in the first step and the pros and cons of the existing systems are identified, if there are any. This step also involves foreseeing the shortcoming of the proposed solutions. The design step involves clearly defining the expected features and processes of the software which is followed by prototyping, implementation and testing. In agile development processes, defined after the agile manifesto (URL-8), these three are packed into short periods of development phases called *sprints*, which produce usable prototypes in each stage for the customers to test. For this reason, agile development processes were taken as an example in proposing the methodology of this research.

1.11. The Process

First of all, the problems faced in the distance education period will be identified. This will be done via semi-structured interviews and surveys with the students and tutors of the Faculty of Architecture. The participants will be expected to have been active during this period. All of the students from all of the departments under the Faculty of Architecture from all years will be invited to participate in the survey. The interviews will be done separately for the tutors and students of the same studios in order to see both sides of the issues.

Secondly, the requirements of the users on both sides will be extracted from the initial investigation. The problems identified will be translated into requirements.

Thirdly, the research in this field and existing software such as virtual environments and video games will be examined to determine the state of the developments in this field. A literature review will be done on the use of virtual environments, video games and game engines in education and architectural design studios. Furthermore, research on distance education and remote working will be examined.

The fourth step will be the evaluation of the state-of-the-art. It will be discussed whether the previous research and the existing software can respond to the identified requirements. Additionally, certain features and methods employed in the discussed research and software will be examined through a semi-technical point of view, in regards to usability in architectural design studios, either in distance working and education, or hybrid conditions.

The fifth step will be the filtering of the requirements by their technical surmountability. The requirements and the technologies at hand will be examined further to find correspondence.

The sixth step is the expansion of the feature list. The possibilities beyond the requirements will be examined and the ones assumed potentially useful to architecture and the architectural design studio setting will be identified and added to the feature list.

The seventh step is the technical design of the proposed virtual environment, as in matching technical/programming possibilities with features to establish a *feature list*. This feature list primarily will consist of technical features corresponding to the requirements. The items will mostly be related to game engines and programming, and will be in technical jargon.

The eighth step is the prototyping, testing and the feedback loop of the developed or proposed virtual environment. The test environment will be the senior year Architectural Design Studio classes, ARCH401 and ARCH402, which will be explained further in detail. Feedback will be collected through observations, surveys and verbal discussions. This step will be repeated as many times as possible. Proposed features will be tested and revisited upon feedback, new features will be added depending on requests and observations.

2. LITERATURE REVIEW

Chronologically speaking, following the preliminary studies, an examination of the literature on various topics related to the research and the findings was done. The following topics and keywords were determined to narrow the examination.

- Virtual Environments
- Multi-User Virtual Environments
- Virtual Collaboration
- Virtual Reality
- Game Engines
- Video Games

There are technological and pedagogical ideas and experimental studies on the use of virtual environments and distance education technologies in architecture. In these studies, various software developed for different purposes were used, add-ons to existing software were developed depending on the scope of the studies, or software prototypes were produced on various platforms such as game engines.

The technologies used and the approaches to the technologies used differ in these studies. These approaches can be summarized in several categories.

Using an existing virtual environment in architecture, developed for a different purpose, is the most common of these approaches. In some of these studies, popular virtual environments and video games such as Second Life and Minecraft have been repurposed to provide educational environments. It is also worth to note that Middle Eastern Technical University has a virtual campus in Second Life (Bulu, 2011) and due to the frequent use in such settings, Minecraft features an educational edition (URL-9).

Aside from using existing software, numerous studies have been made on the use of game engines in architecture. The first problem faced in this approach is the conversion and transfer process (pipeline) of the commonly used data formats in architectural CAD software to game engines. Studies have been made on this aspect of the use of game engines in architecture and recently more frequently on the use of VR technologies in architecture, which is also possible through the use of game engines. Studies have also been conducted on the use of AR technologies provided by game engines in architecture, which requires the same steps to be taken before additional processes involving AR.

After being able to import architectural drawings, or models in most cases, numerous studies have been made on the use of real-time rendering capabilities in game engines. Real-time rendering capabilities often provide support for real-time interaction and roaming inside these models in full detail the engine and the accompanying features provide.

Not only limited to rendering and visualization, game engines also provide various interaction, analysis, simulation and computation capabilities on these imported models. Separating real-time environments that game engines offer from conventional CAD environments, is the ability to see continuous changes, *in real-time*. This has been the focus of some of the examined research.

Moving on to one of the more complex features of the game engines, numerous studies have been made on the multi-user capabilities and collaboration possibilities through these capabilities. This topic can be considered more complex due to the fact that it requires networked programming and compiling of the code to provide actual support for multiple users. These capabilities are often tied with communication and collaborative design possibilities.

In conclusion, the aforementioned approaches can be listed into research topics as follows:

- Use of existing virtual environment software
- Transfer of data types commonly used in architectural CAD software to game engines
- Investigation of visualization possibilities of architectural models transferred to game engines
- The use of various interaction possibilities of game engines

- Use of game engines as computation, analysis and simulation tools
- The use of virtual reality and augmented reality tools through game engines
- The use of multi-user virtual environments in communication
- The use of multi-user virtual environments as collaborative design tools

These topics can be detailed further on whether they are focused on architectural design, architectural education or AEC industry in general. The topics were defined in the most general sense to avoid cluttering, as they were determined only to point to some of the many possibilities of the use of game engines in architecture whether it is for architectural design or education.

Gül (2020) summarizes the history of the use of computer, computational design and virtual environment technologies in the field of architecture and the possibilities of using these technologies in the future. They state that with the widespread use of computers, a computational thinking-oriented basis has emerged in the fields of architectural production, application and design. The inform that with the introduction of affordable personal computers since the 1990s, the use of computer aided drawing programs in the field of design has increased, and since the 2000s, analysis, simulation, structural and formal form-finding methods have begun to be carried out in the computer environment with the architectural design-oriented software developed in this field. They argue that curvilinear and parametric surfaces have become popular because of the ease of use of the tools used, in the context of the tools used in the design being the determining factors of the limits and possibilities of the design. They also state that in the same period, popularized virtual environment software was tried in many architecture schools and architectural design studios. Having researched the potential of multi-user virtual environment software such as "Active Worlds" and "Second Life" in the field of architectural design, Gül emphasizes the importance of interaction. They state that in order to create a convincing, realistic and tangible experience, an uninterrupted and natural human-computer interface is required, and this interface can be strengthened by the emerging virtual reality technologies. According to Gül, the most important features of the use of virtual environment technologies in architectural design are that they enable interaction and provide an experience-oriented environment; Based on this, they argue that virtual environment software to be used in architectural design should focus on experience as well as information processing and computation capabilities.

Grasser et al. (2020) explore real-time collaborative design possibilities in architecture with their multi-user and cross-platform supported application. Developed with the Unity game engine, the software makes use of augmented reality technologies on mobile devices and can be used simultaneously in the computer environment with participants from different platforms. They also tested remote collaborative design scenarios with software that includes text-based communication functions. Grasser et al. reports 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 that the most popular virtual environments are those that support user-created content, and that collective creativity and knowledge sharing arise from this content. They state that collective creativity and collaborative design can cause problems in terms of authorship, but the main purpose of this type of work is to provide a common output. Grasser et al. report that collaborative environments can support multiple goals and views, as well as exploration and experimentation in design where real-time interaction and feedback are possible. In their studies, they stated that the use of more than one software brought difficulties due to the workflow layout.

Chien et al. (2020) presents a workflow proposal that aims to provide an efficient, simple and fast connection (pipeline) between CAD software and VR hardware. They state that VR-supported virtual environments are mostly research-oriented and not easily used by architects, and existing software suitable for use in the field of architectural design is also not accessible due to reasons such as high fees. Chien et al. describe in technical detail an easily imitated workflow proposal. Stating that the format and geometric features of the models transferred from the CAD environment to the VR environment, exported from CAD software, are not suitable for VR, they divide the methods that enable the transition between environments into three. The "Conversion" method is one-way, the model transferred from the CAD environment to the VR environment is suitable for use for presentation purposes, and it is not appropriate to use this method during the design phase. The "Link" method is bidirectional and relies on the data lost during the conversion process to be stored with special methods and then read and matched with complex methods during the conversion stages. It is possible to experience data loss and has various difficulties. The "System" method, on the other hand, is based on the development of a VR-based CAD environment. Chien et al., who argue that the difficulties of trying to design in a VR environment and that architects can work more effectively in the CAD environment, state that this method is also not suitable. Expressing that the most appropriate use of VR technologies in the field of architectural design is in the field of communication and participatory design, Chien et al. propose a scheme that will support these fields. The proposed scheme is based on editing the Rhino model with Grasshopper scripts and making it suitable for use in the VR environment. Instead of designing in a VR environment, their proposed approach is based on programming the variable/parametric components of the design in a way that allows manipulation in the VR environment by separating it from the rest of the model. Interactive objects are programmed in the Unity environment and the fixed models are combined and transferred to the VRChat virtual environment and experienced in the VR environment. The changes and design decisions made are saved in the JSON (Javascript Object Notation) format, which can be used to transfer data between most programming languages and software, and are read with the help of Grasshopper scripts and applied to the Rhino model. With this method, interactive game objects created as "Prefabs" in the Unity environment can be manipulated, combined and separated by operations such as changing location, direction and size in the VR environment, and objects programmed to include parametric and/or variation can be easily transformed.

Hong et al. (2019) explores the possibilities of using “Multi-User Virtual Environments” in architectural design in their research. Their research focuses on the potential of these environments for creative collaboration with remote access. The researchers tested the use of "Second Life" and "Groupboard" software in this context and evaluated the design results of the participants with the help of the "Consensus Assessment Technique" (CAT). Hong et al. argue that the environment to be used in creative collaboration forms the basis of communication between designers, and that the environment to be used in collaboration in architectural design should reflect the three-dimensional form and volume, performance and usability of the building. They state that too abstracted forms of representation will hinder communication and cause communication problems due to misinterpretation. They state that multi-user virtual environments are an alternative to physical coexistence, and that the communication tools offered by these environments support creative collaboration. They describe that the immersion made possible by the mentioned environments provides the feeling of being in the design environment, and the coexistence of the users in the virtual environment with the help of "avatars" strengthens the sense of presence and creates interaction opportunities. They state that another effect of using avatars is to be able to understand the physical and functional features of the design. They also state that simultaneous and shared digital objects in the virtual environment contribute to the sense of presence and the environmental atmosphere of the virtual environment. As a result of their experiments, it was revealed that designs made in multi-user virtual environments were evaluated as more “innovative” and “appropriate” than those made in online drawing software. Researchers think that this result is due to the use of avatars and co-existence in multi-user virtual environments, which strengthen communication and collaboration between designers. According to the observations of the researchers, in these environments that contribute to the sharing and experience of spatial information, the participants had the opportunity to discover their design decisions with the help of their avatars and to experience their physical features in the digital environment. As a result of the research, Hong et al. state that multi-user virtual environments are suitable for use in ergonomic, human-scale design projects and design performance analysis related to the activities of users in these projects, and they can be used in collective and qualitative evaluation methods in remote collaborative design studies.

Sandstrom and Park (2019) developed a shape grammar game on spatial constructs in architectural design. The aim of the developed software is to obtain rules from the created spatial constructs instead of creating spatial constructs in the context of rules. Developed with the Unity game engine, the software monitors the decision-making mechanisms by recording the movements of the users, and offers data that can be analyzed about the decision-making methods of the user from the data recorded when the game is over. This study sets an example in deducing the usage statistics of users in the field of architectural design, such as recording usage data and analyzing the design decision mechanisms of designers.

Leitao et al. (2019) compare the potential of use of game engines in the field of architectural design with architectural visualization engines within the scope of their visualization capabilities. Within the scope of the research, the Unity game engine and the CineRender visualization plug-in of the ArchiCAD software were compared in terms of navigability, the time required per visualized frame and the image quality of the produced image. Leitão et al. states that the increasingly complex designs, thanks to the increasing capabilities of current architectural design tools, can no longer be expressed with two-dimensional representation methods. For this reason, while talking about the importance of three-dimensional visualization tools, which have become the industry standard in the field of architecture, they mention the deficiencies of three-dimensional modeling/CAD software in visualization. They emphasize that plug-ins and stand-alone software developed to make up for the visualization deficiency of modeling software require long periods of time to create images, and therefore software that will provide intervention and real-time visualization opportunities is needed. They argue that game engines that offer real-time visualization and interaction can meet this need, but they also discuss that transferring architectural models to game engines is often a problematic process. They state that digital modeling in the field of architecture can be divided into two as BIM and CAD, and because CAD models contain only visual information, they can be transferred to game engines more easily than BIM models that contain metadata. The advantages of using the Unity game engine in architectural visualization, which the researchers mentioned in detail: The advantages of the Unity game engine in architectural visualization can be summarized as: the ease of integration of current technologies such as VR, the programmability of features such as three-dimensional sound and climate/season simulations, the real-time visualization capabilities, the use of time in the early stages of the design, and the more realistic navigation, which the researchers mentioned in detail. The most noticeable difference in the experiments performed is in the time required per visualized frame. They completed the same length part of the video sequence with Unity in 9 minutes, which resulted in an error with CineRender at the 470th hour. According to the results obtained from the study, the visual quality of the game engines with optimized performance for games decreases, but the visualization quality remains the same even when roaming, and no waiting is required for the visualization. The advantages of using the Unity game engine instead of the CineRender-like visualization engines are stated as: seamless visualization, real-time texture, light and shadow visualization, realistic interaction and navigation, VR support, and performance/quality adjustments depending on the situation. Another feature that can be realized in game engines, mentioned in the research but not implemented, is wheelchair accessibility simulation.

Gül (2019) conducted a study investigating the effects of tools used on designers' behavior in real-time collaborative design. Emphasizing that designers communicate with representation tools, Gül states that representation tools, which provide various cognitive and interactive tools, also affect the behavior of designers. They also state that studies on the impact of virtual environment technologies on designers in terms of cognition, communication and interaction are out-of-date as a result of the perpetual evolution. They state that virtual worlds imitate the real world, but they provide a more suitable environment for design by removing physical restrictions and thus strengthening creativity. Gül explains that "Second Life", one of these virtual worlds, is suitable for design as it is independent of game mechanics, offers modeling opportunities and has sufficient graphic quality. Within the scope of the study, a mobile augmented reality software was developed with the Unity game engine and tested against pen and paper sketching around the table, collaborative digital sketching over the internet (Groupboard) and a multi-user virtual environment software (Second Life). The tests were evaluated with the protocol analysis method, and it was revealed that two-dimensional environments triggered discussions on behavioral and functional design criteria, and three-dimensional environments on structural design criteria. Another behavior observed in three-dimensional environments is the use of time focused on making, with most focus on editing objects. Regardless of the technology used, it has been evaluated that being in the same physical environment triggers collective work, and remote cooperation triggers individual work. The reason why the use of the virtual world triggers this situation is explained as the fact that individuals have their own camera perspectives and can act independently in these environments. Another important output of the study is the effect of observing the actions performed on the design process carried out in virtual environments. Design actions that are not communicated in real-time in virtual environments cause communication problems. In remote collaboration situations without real-time action awareness, planning and division of labor are required beforehand. It has been noted that in the cases where people have the same perspective, described as "what you see is what I see" by the author, the planning can be done more successfully.

In their research, Bartosh and Philip (2019) examine the potentials of using VR technologies as a tool in architectural education. The focus of the research is the use of VR technologies in the early design phase for information visualization, dynamic interaction, interactive learning and communication. Arguing that VR technologies may be the next step in using digital technology in architecture, Bartosh and Philip summarize the advantages of these technologies as simultaneous presence, real-time modeling, three-dimensional data visualization and the use of design tools in the environment. They associate the fact that VR technologies have not yet become widespread in the field of architecture with the limited knowledge of programming among architects, but they state that designers have regained interest in these technologies with the production of affordable VR hardware. Bartosh and Philip mention that VR technologies are generally used in the final stages of design and there are difficulties in transferring CAD models to VR environments. They state that VR technologies can be design and analysis tools, not just representation tools, thanks to their programmable interaction features, but they also mention that modeling the design in the CAD environment and transferring it to the VR environment alleviates the learning curve of VR tools. They discuss the limitations of existing VR software and that the user experience is guided by the capabilities of the tools, emphasizing that these limitations can be overcome with the use of game engines. They state that necessary interactions can be programmed in VR software developed with game engines, and in this context, they have made various experiments with the Unity game engine. They visualized simulations of environmental data such as acoustics, daylight, ventilation in a three-dimensional environment and programmed various interactions. As a result of the experiments, they state that the use of the game engine highlights the experiential features of the design, that the participants prefer the visual interface instead of the voice command, which is within the possibilities of the game engine, and that the use of the VR environment as a design tool allows them to design beyond conventional assumptions.

Pienaru (2018) explores the possibilities of using game mechanics and game technologies on the use of big data in the field of urban design. Pienaru states that the use of urban data, which is open to access but cannot be used by the mainstream user, in a game setup can benefit urban design. Pienaru developed two different games with two different groups of participants in order to use this data. One of the games developed within the scope of the research was produced with the “Processing” IDE (integrated development environment) built for visual arts, and the other with the Unity game engine. Pienaru states that games are environments that can safely simulate design decisions, and that game mechanics allow the use of an unmanageable amount of data.

Nandavar et al. (2018) proposes an open-source workflow prototype between BIM software and VR environment, bidirectional, independent of software and hardware. Arguing that VR tools are tools that will strengthen collaboration and communication beyond visualization, Nandavar et al., within the scope of their research, examined various workflow model studies and current software in the market and explained the shortcomings of these models in detail. As a result of their research, they presented a workflow proposal by making use of the IFC format, which can store the metadata of BIM models, and the Unity game engine. In the prototype they developed, they used various data compiler code modules in the Unity software in order to provide bidirectional data flow, which they used to manipulate BIM objects in the Unity environment and transfer the changes to the BIM environment. Expressing the importance of data visualization in the field of BIM, the authors mentioned the various possibilities provided by VR technologies in this regard. They stated that game engines enable graphics and performance optimizations and interaction, and they used features such as human-scale navigation, distance and area measurement, item querying, conversion and deletion, visual marking, taking screenshots and leaving a voice message in the prototype they developed. They state that the envelopment effect that VR technologies provide by providing representation and presence on a 1:1 scale strengthens remote work, and is efficient in terms of effort, time and resource use in large-scale projects that require a large number of experts. Stating that the use of VR strengthens communication and improves the user experience, the researchers state that they will also investigate the multi-user potentials brought by game engines in their future studies.

Sorguc et al. (2017) investigate the potential of using VR technologies in architectural education. The research was conducted within the scope of the "Digital Design Studio" courses at the Middle East Technical University. Stating that VR and related technologies can create their own reality rather than imitating the real world, the researchers argued that VR is a new space and experience environment and carried out their studies within this framework. According to the research, VR experiences trigger at least three types of learning methods, primarily visual, auditory and kinesthetic. Researchers, who state that design studios should be questioned in the context of developing technologies, mention that courses with computational design in architecture conducted in this direction are doubly difficult. In such scenarios, researchers state that while teaching design, the use of related technologies should be taught, they also argue that the limits and potentials of the environments and tools used should be recognized and these limits should be pushed. It is understood from the various terms used by the researchers, who do not share the technical details of the VR environment they use, and the screenshots they publish, that they use the Unity game engine. Stating that in their studies, abstract, incomplete and unrealistic problems produce better results in computational design studios, the researchers also stated that the participants instinctively imitate the real world while designing a virtual world.

Moleta (2017) explores the potentials of dynamic environment simulation in architectural visualization in their architectural design studios. Focusing on supporting the virtual spaces produced in virtual environments with dynamic environmental elements, Moleta argues that such interactions will improve the user experience and strengthen the immersion potential of the virtual environment. They observed that the increase in the level of detail of VR environments produced within the scope of architectural visualization caused the users to focus on visiting the environment rather than the buildings. In detailed and large virtual environments, users navigate until they see everything, and when they see everything or reach the limits of the virtual environment, they feel like the "game" is over. They state that photorealistic virtual environments are possible with today's technologies, but photorealism does not allow us to understand the relationship of the buildings with dynamic environmental conditions. He argues that virtual immersion is not possible with photorealism alone, and that although the photorealistic features of game engines are weaker due to performance concerns, the dynamic features they offer further strengthen immersion. Regarding the environments used in architectural visualization, they state that the models examined by moving rapidly due to the usage practices of CAD environments do not reflect the experiential features of the design. They argue that the static frames generally used in visualization techniques cannot reflect architecture, but game engines are also motion-oriented and are not directly suitable for use in architectural visualization. Describing that VR-supported game engines can reflect the architecture, but experience-oriented tools should be used, Moleta states that it is not necessary to produce these tools and that they are already in game engines. In this context, they used tools that simulate dynamic environmental conditions. Moleta preferred Unreal Engine due to its ease of use, but stated that they would prefer Unity in their future works. Moleta, who first left the environmental conditions to the user's control with the software they developed within the scope of the research, later commented that it is beneficial for game engines to provide this dynamic interaction, but it is more interesting to control this interaction by real-time data. On top of that, it used real-time climate data from “Yahoo Weather” and “WUnderground” platforms for architectural visualization. They state that, thanks to the architectural visualization made with real-time data, unlike the act of movement supported by game engines, the participants watched the variable environmental conditions by standing at one point, and even re-entered them on different days and at different times of the day to check them.

Black and Forwood (2017) explain the potentials of using game engines in engineering simulations in their research. Focusing on the interactive visualization of complex structural analyses on the game engine platform, the communication possibilities brought by real-time data visualization are emphasized. In their study, the Unity game engine, which provides real-time visualization, was used in order to visualize the structural analysis of a moving facade design during movement. Expressing that complex structural behaviors such as moving facades cannot be analyzed with static calculation methods, Black and Forwood state that such situations require an intense and repetitive process with conventional methods. They state that using a game engine in engineering calculations enables real-time data visualization, and that the developed software can be used without requiring any additional software or programming knowledge, and can be distributed in compiled .EXE or even web-based WebGL format.

Moleta (2016) researches the use of real-time virtual environments in the digital design workshops they conduct and reports the studies done within the scope of the course. They state that with the inclusion of game mechanics in the virtual environment, these environments can be used beyond visualization. Stating that game engines have been used by architecture students for a while, but is limited to use for visualization purposes, Moleta states that this is due to the difficulties of using game engines, and therefore they are only used in the final stage of the design. They state that with the use of game engine platforms in design studios, they can also be used at the initial stages of the design, and that game mechanics will contribute to the active participation of students in design studios. While they state that simulation and scaled models cannot fully provide a sense of presence, they describe that game engines can also be used to express the experiential aspects of the design. In this context, they state that game engine technologies can fill the gap between the design and building processes by making it possible to experience the design before it is built. As a result of the studies, it is stated that students can better understand human scale, dimensions and distances thanks to the use of first-person perspective and navigation possibilities. It has been observed that designing experiences through game mechanics has various behavioral effects on students. Even in architectural design studios, students who tend to display protective behaviors about their designs and hide them from others tend to share the projects they have produced within the scope of this course, receive feedback, and develop their projects in line with the feedback. As a result of these observations, it was noticed that the students perceived that the ultimate purpose of their designs was the use of people other than themselves and they were more active in the studio environment. Moleta states that the use of game engine technologies in design studios encourages students to think more user-oriented and makes it possible to experience their designs from the inside rather than observing them from the outside.

Du et al. (2016) draws attention to the lack of multi-user VR environments in the AEC industry, and develops and tests a software that can be used in this field with the help of the Unity game engine. Focusing on communication and collaboration, the study states that multi-user VR environments can support the project planning and programming of the involved parties in the early stages of construction. A game engine-based software, which includes interactive navigation, marking and voice communication tools, was produced by editing BIM models and transferring them to the game engine platform and compiling the software on a model basis. Researchers also mention the possibility of BIM supported, game engine-based software to perform emergency evacuation simulations.

Vals et al. (2016) convey their experiences in using game engines in architectural design studios. Their work focuses on the architectural potential of the simulation and gamification (first-person perspective) possibilities of game engines. Stating that architecture and computer games are always related, Valls et al. state that this relationship starts with the architectural elements forming the background for the level designs and then takes it to the center with the city building games. They state that agent-based simulations used in city building games can be used as city planning tools, and they state that virtual spaces created with geospatial data are currently used in geographic analysis. They also mention that realistic environment simulations developed with game engines are used as decision-making tools in collaborative design in the field of landscape architecture. They discuss that innovations such as the emergence of CAD software affect the form of architecture, and they state that environmental simulations can have similar effects. Researchers preferred to use the Unreal Engine because less programming knowledge is needed in its basic use and stated that they could switch to the Unity game engine in their next studies. They state that during the development phase of this type of software, the feedback loop should be kept short in order to meet the needs of the users and avoid unnecessary developments. In their study, they state that the use of first-person perspective and navigation, instead of the pan and orbit controls that are usually used in architectural software, strengthens the sense of being in space and the perception of human scale. They state that usage data of users' browsing and browsing can be collected with the help of game technologies, and they state that they plan to implement it in their future research.

Yan and Liu (2007) conducted a gamified architectural software development research that offers simulations of sustainability and design performance, which can be used in architectural design practice and education. The study, which connects the building information modeling software Revit and the Microsoft XNA game engine, focuses on areas such as usage simulations of various user profiles, resource management, performance analysis based on building material, and budget management. Although the software and software versions used are out of date, the various methods used in the study with the agent-based design and analysis tools carried out by using calculation methods such as Dijkstra's shortest path algorithm remain up-to-date.

3. PRELIMINARY STUDIES

In order to identify the problems faced during the distance education period, a student survey and two semi-structured interviews, one with the students and the other with the studio tutors, were conducted. The aim of these studies was to identify the problems and advantages of the current distance education methods through listening to both sides of the communicating parties.

The sequence of these studies were planned in an order that would provide additional discussion points to later studies, as in the student survey would define further topics for the tutor interview and the discussions held in the tutor interview would contribute to the topics to be discussed in the student interview.

These preliminary studies were done to identify the initial, basic, foundational problems and advantages of the current methods. The discussion of these identified points, when approached from a more technical point of view, were expected to define the underlying factors thus providing the fundamental items for a *requirements list*.

3.1. Student Survey

First of all, a survey for the students was prepared through speculative discussion and initial examination of the studio recordings of the first term of distance education. The questions were prepared in order to find the underlying relationships between the problems faced and the design, presentation and communication methods and tools used by the students and the tutors. Some questions in the survey were prepared to provide hints for possible solutions through asking about control schemes, software preferences and the usage frequency of various methods utilized in used software.

The types of the questions in the survey vary. Scales, multiple choices and text boxes were used in collecting responses for different questions. Microsoft Office Forms was used to create the survey and collect the responses. Users were prompted to login with their school emails to ensure one response per person but the identities of the respondents weren't paired with the collected responses.

One advantage besides being able to collect one response per person in using Office Forms was the "Insights" the application can provide. Without manual comparison or the need to use additional analysis software, Office Forms can analyze and provide insights on the responses depending on automatically analyzed correlation.

All students from Istanbul Bilgi University's Faculty of Architecture were invited to participate in the survey, under the condition of having been active in the distance education period. Therefore, the questions were prepared to be less architecture-oriented and more inclusive of the various disciplines, such as Industrial Design, of the Faculty of Architecture. 33 responses were recorded anonymously.

The questions, the aim of the questions and the recorded responses are as follows:

Question 1:
Which of the following design studio courses have you attended in the current semester?

This question was asked in order to analyze responses of the following questions in context to student experience level. Response correspondence to various disciplines and experience levels was generated after all the responses were collected through the use of the answers to this question.

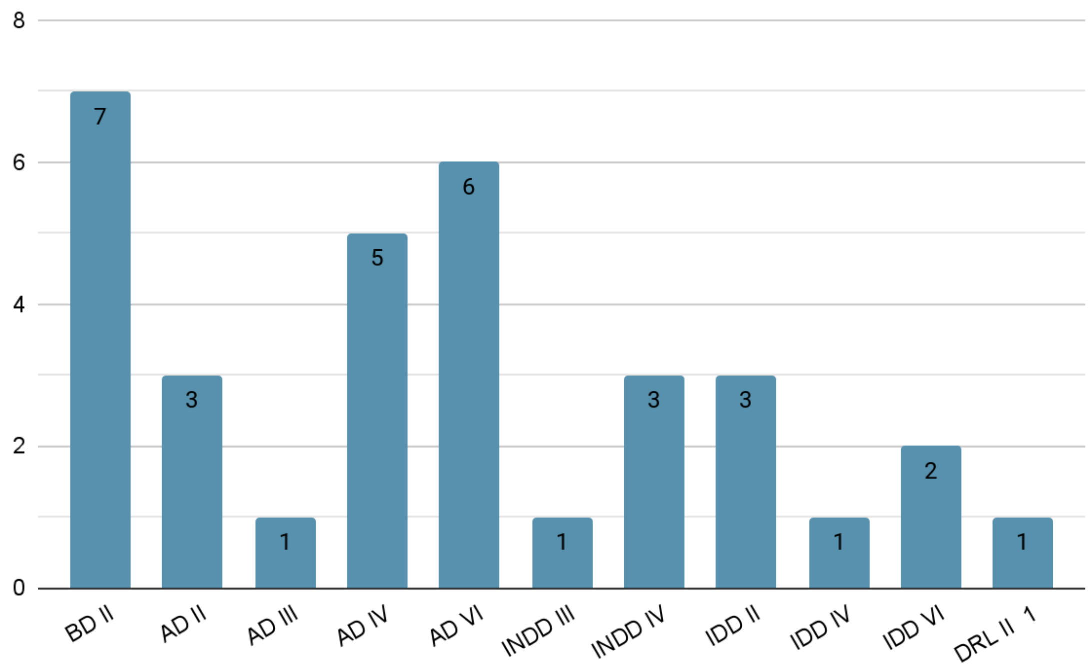

Figure 3.1. Answers for Question 1

Answers for question 1:

- Basic Design I 0
- Basic Design II 7
- Architectural Design I 0
- Architectural Design II 3
- Architectural Design III 1
- Architectural Design IV 5
- Architectural Design V 0
- Architectural Design VI 6
- Industrial Design Studio I 0
- Industrial Design Studio II 0
- Industrial Design Studio III 1
- Industrial Design Studio IV 3
- Industrial Design Studio V 0
- Industrial Design Studio VI 0
- Interior Design Studio I 0
- Interior Design Studio II 3

● Interior Design Studio III	0
● Interior Design Studio IV	1
● Interior Design Studio V	0
● Interior Design Studio VI	2
● Design Research Lab I	0
● Design Research Lab II	1
● Design Research Lab III	0

Basic Design I and II are common design studios for the first year students of the departments of Architecture, Interior Architecture and Industrial Design. Design Research Labs are graduate-level design studios. The question was asked in a way that accepted multiple answers due to the fact that students can enroll in multiple design studios in the same term.

Question 2:
Which of the following CAD software do you use?

This question was asked to determine one of the technical requirements, the data format(s) that should be supported by the environment.

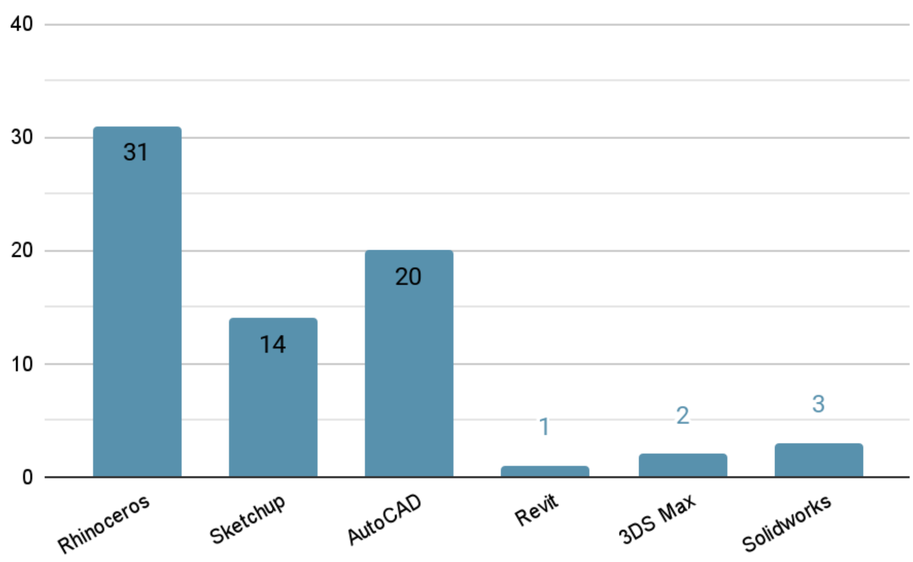

Figure 3.2. Answers for Question 2

Answers for question 2:

- Rhinoceros 31
- Sketchup 14
- AutoCAD 20
- Revit 1
- 3DS Max 2
- Maya 0
- Blender 0
- ArchiCAD 0
- Other (5)
    - Solidworks 3
    - Lumion 1
    - VRay 1

The question specifically asks for CAD software, the answers "Lumion" and "VRay" were disregarded as those are software only capable of rendering 3D scenes, not drawing.

Question 3:
How comfortable are you with using pan and orbit controls while navigating through a digital environment?

This question was asked to determine the proficiency of the respondents' in using the control schemes commonly used in CAD software. Pan and orbit controls usually consist of mouse controls combined with keyboard modifiers to provide the ability to pan the view and orbit around objects. The mouse cursor is usually free in CAD software, when a key or a mouse button is held, the movement of the mouse is translated to either pan or orbit originating from the mouse location. Camera roll can usually be achieved through key combinations. This question was asked on a scale from 1 to 5.

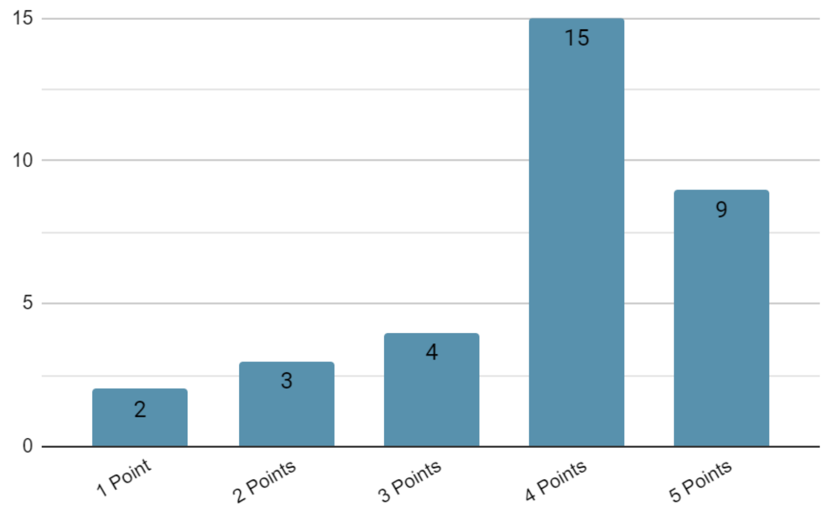

Figure 3.3. Answers for Question 3

Answers for question 3:

- 1 Point 2
- 2 Points 3
- 3 Points 4
- 4 Points 15
- 5 Points 9

Average Rating: 3.79

Question 4:

How comfortable are you with using a mouse + keyboard control scheme in first person perspective (WASD) while navigating through a digital environment? (e.g. Lumion and FPS video games)

Similar to the previous question, this question was asked to determine the proficiency of the respondents' in using the control schemes commonly used in first-person and third-person video games. These control schemes vary in camera controls, but usually consist of using W, A, S and D keys to control a character in the horizontal plane. Camera controls can differ based on the camera perspective and the game genre. FPS (first-person shooter) and TPS (third-person shooter) games usually lock the mouse to the center of the screen where it becomes a crosshair, in such schemes any mouse movement is directly translated to the rotation of the camera in terms of pitch and yaw. Camera roll is uncommon in such camera control schemes. This question was asked on a scale from 1 to 5.

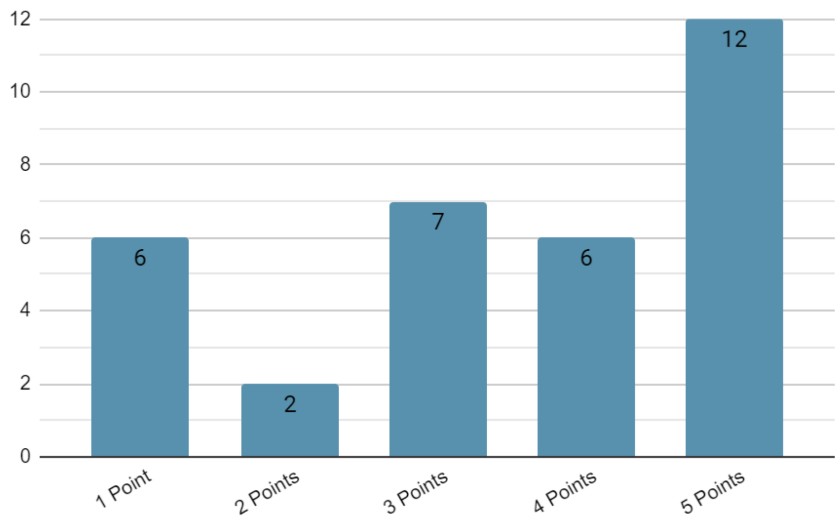

Figure 3.4. Answers for Question 4

Answers for question 4:

- 1 Point 6
- 2 Points 2
- 3 Points 7
- 4 Points 6
- 5 Points 12

Average Rating: 3.48

Insights for question 4:

Out of the people who answered “5 Points” for the previous question, %89 of them answered “5 Points” for this question as well.

Question 5:

Which remote learning platforms and services have you used?

This question was asked to find out correlations between the problems faced and the platforms used. While all the platforms used in distance education methods can be described as online communication tools, they vary in collaboration capabilities through the tools they provide. While some applications focus on online video conferencing, others may focus on online collaboration through drawing and resource sharing, such as text, images and videos on an online virtual whiteboard. Some of these software can also provide screen sharing, remote control and drawing on top of the shared screen video stream.

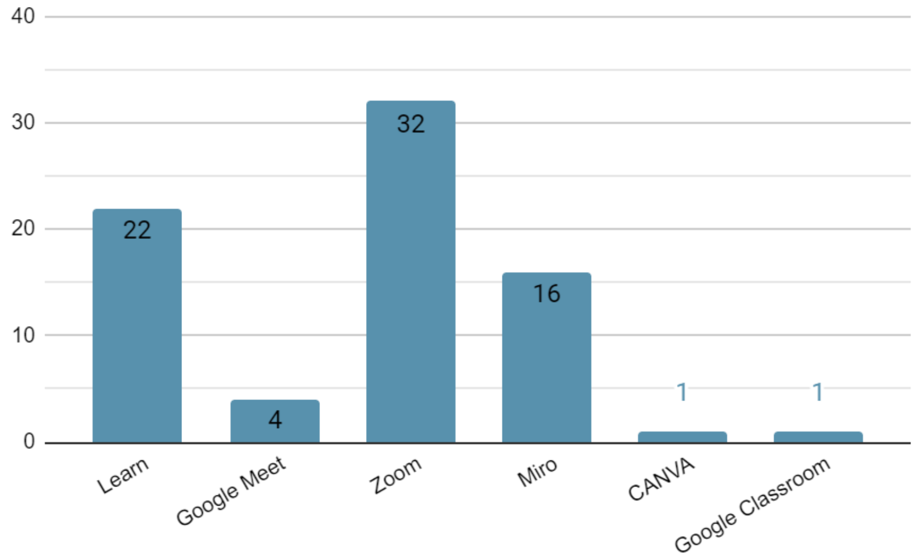

Figure 3.5. Answers for Question 5

Blackboard is Istanbul Bilgi University's in-house online application for course tracking. This web-based application features course scheduling, online submission capabilities and tutor announcements. During the distance education period, this platform was used to announce video conference links, distribute course materials, collect student submissions and perform online exams and quizzes.

Google Meet is a video conferencing app developed by Google. Google Meet is web-based, which means it doesn't require any installation by the users. It features video, voice and text communication and screen sharing. There are no drawing capabilities on top of the screen sharing video stream but it features a collaborative whiteboard application where users can draw and upload images within meetings.

Microsoft Teams is a remote working and collaboration platform developed by Microsoft. Teams can be used either on an internet browser or through the desktop client. It has various features depending on the selected template. For distance education, Teams features video conferencing, individual permanent text chat rooms, announcement and course scheduling capabilities. Screen sharing and drawing either on the video stream or a blank page is possible. Taking remote control is also featured while a user is sharing their screen. Teams also features capabilities for online submission, file sharing, holding online exams and quizzes which tutors can grade inside the app either manually or automatically depending on the type of the questions. Tutors can also generate attendance lists for the classes held online.

Zoom is a video conferencing application which runs only on an executable client, whether it is the mobile application or the desktop client. Zoom features online video, voice and text communication and file sharing inside meetings. It also features screen sharing, taking remote control of the screen sharing users' device and drawing on shared screen video stream. Users can also create online whiteboards inside and outside of meetings to collaboratively work, which features text boxes, various drawing tools and image uploading capabilities.

Miro is an online collaboration whiteboard application. It has similar features to Zoom's and Google Meet's whiteboard applications. Users can write text, draw shapes and diagrams through various drawing tools and upload images and videos into the online whiteboard. Miro works on browsers but also features mobile and desktop clients.

Answers for question 5:

- Blackboard (Learn) 22
- Google Meet 4
- Microsoft Teams 0
- Zoom 32

- Miro 16
- Other (4)
    - CANVA 1
    - Google Classroom 1
    - Drive 1
    - Youtube 1

Several responses outside of the given choices were given to this question, and the answers were examined in terms of tool capabilities.

Canva is an online whiteboard tool similar to Miro but is graphic design oriented, it could be summarized as an online, collaborative graphic design tool similar to Adobe InDesign with some image editing capabilities similar to Adobe Photoshop.

Google Classroom is Google's online education platform similar to Blackboard and Microsoft Teams' classroom template but limited in individual online communication capabilities. It features course scheduling, online submission, exam and quiz capabilities with online grading. It is mainly focused on online submissions, examinations and grading.

Drive is Google's cloud storage solution. It was not regarded as a distance education platform in this research.

Considering Youtube as a remote learning platform was found interesting but this statement has no concernable addition in the scope of this research.

Question 6:

For how long can you stay concentrated on a remote learning session?

This question was asked to find the attention span of the students' in online education. The correlation between the problems faced and students' attention span was also sought through the responses to this question.

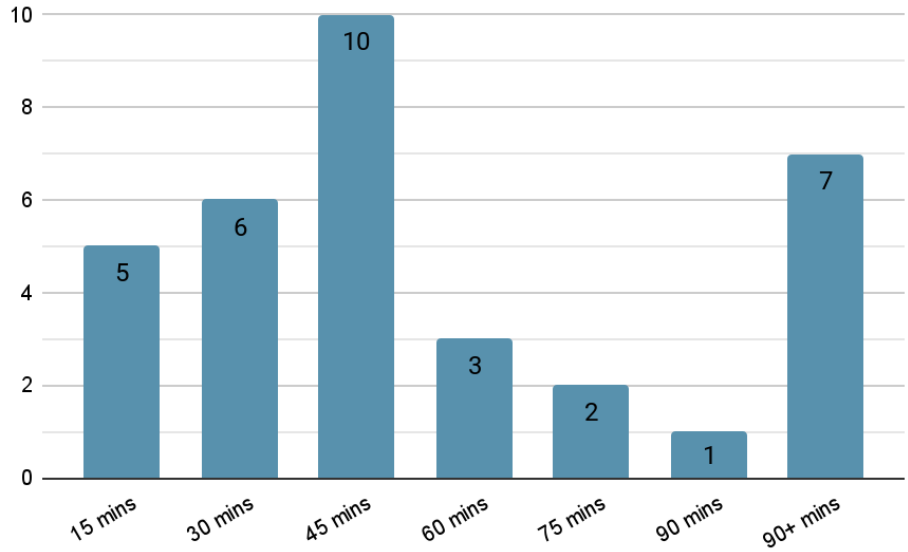

Figure 3.6. Answers for Question 6

Answers for question 6:

• 15 minutes	5
• 30 minutes	6
• 45 minutes	10
• 60 minutes	3
• 75 minutes	2
• 90 minutes	1
• More than 90 minutes	7

Questions 7-10

The following questions were asked to determine the drawing methods and the media used in various stages of design. These questions were asked to find out if certain drawing methods were preferred in different stages and if certain drawing methods and media are preferred over others. The outcome of the responses were thought to be capable of pointing to certain requirements in particular stages of design.

Question 7:

Which of the following drawing methods have you utilized while designing your project?

This was a multiple-choice question, inquiring about all the drawing methods used.

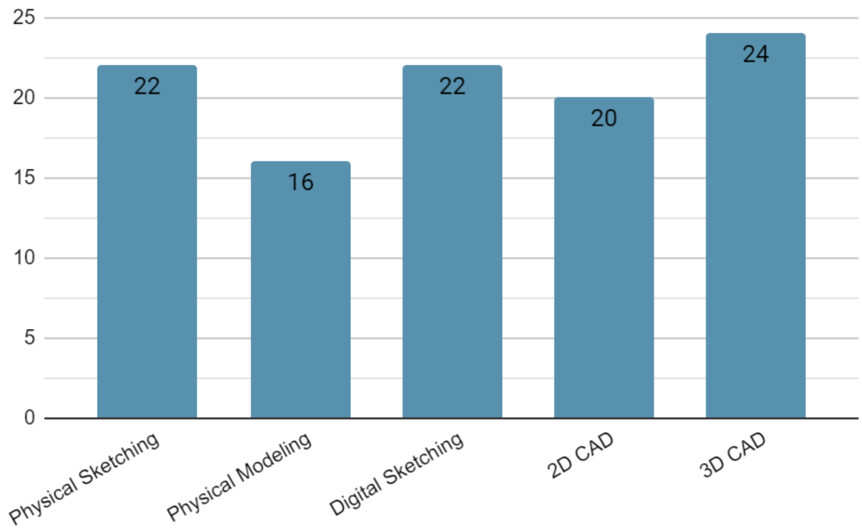

Figure 3.7. Answers for Question 7

Answers for question 7:

- Physical Sketching 22
- Physical Modeling 16
- Digital Sketching 21
- 2D CAD 20
- 3D CAD 23
- Other (2)
    - Photoshop 1
    - 3D Modeling 1

The single, custom response of "Photoshop" was regarded as digital sketching, and similarly "3D Modeling" was counted as "3D CAD".

Question 8:

Which of the following drawing methods have you utilized the <u>most</u> while designing your project?

This was a single-choice question, inquiring about the most used drawing method.

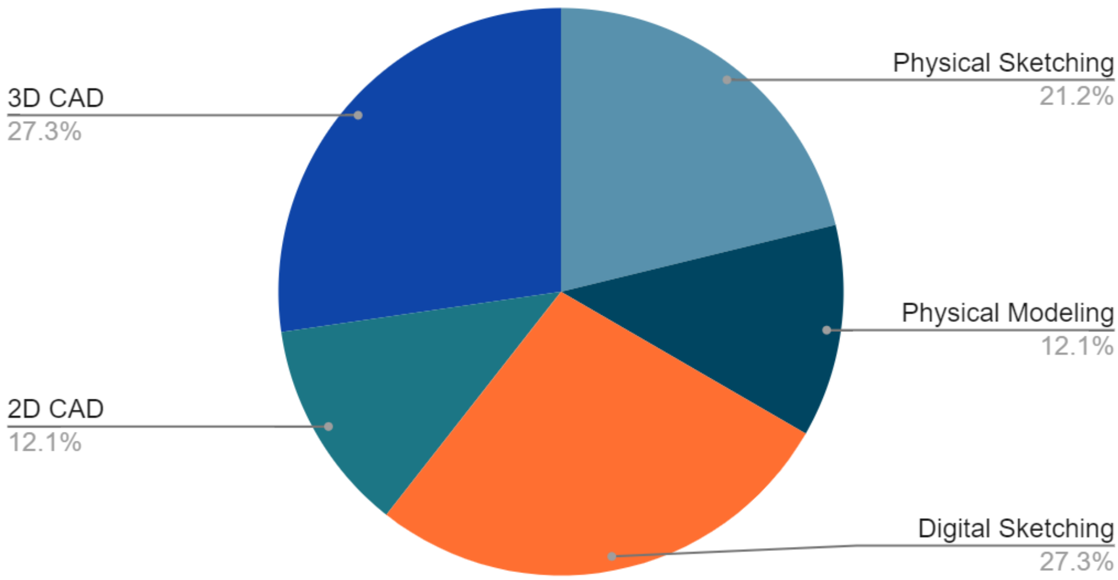

Figure 3.8. Answers for Question 8

Answers for question 8:

- Physical Sketching 7
- Physical Modeling 4
- Digital Sketching 8
- 2D CAD 4
- 3D CAD 9
- Other (1)
    - Photoshop 1

Question 9:
Which of the following drawing methods have you used <u>in the early stages</u> of the design?

This was a multiple-choice question, inquiring about all drawing methods used in the early stages of design.

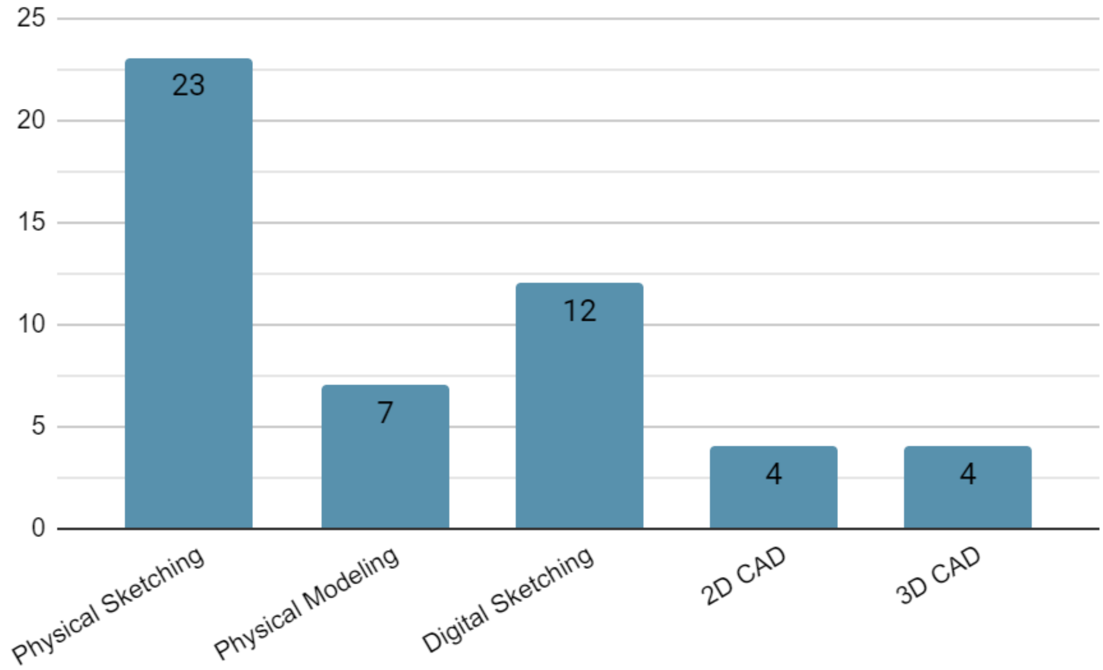

Figure 3.9. Answers for Question 9

Answers for question 9:

- Physical Sketching 23
- Physical Modeling 7
- Digital Sketching 12
- 2D CAD 4
- 3D CAD 4
- Other 0

Question 10:
Which of the following drawing methods have you used <u>in the final stages</u> of the design?

This was a multiple-choice question, inquiring about all drawing methods used in the final stages of design.

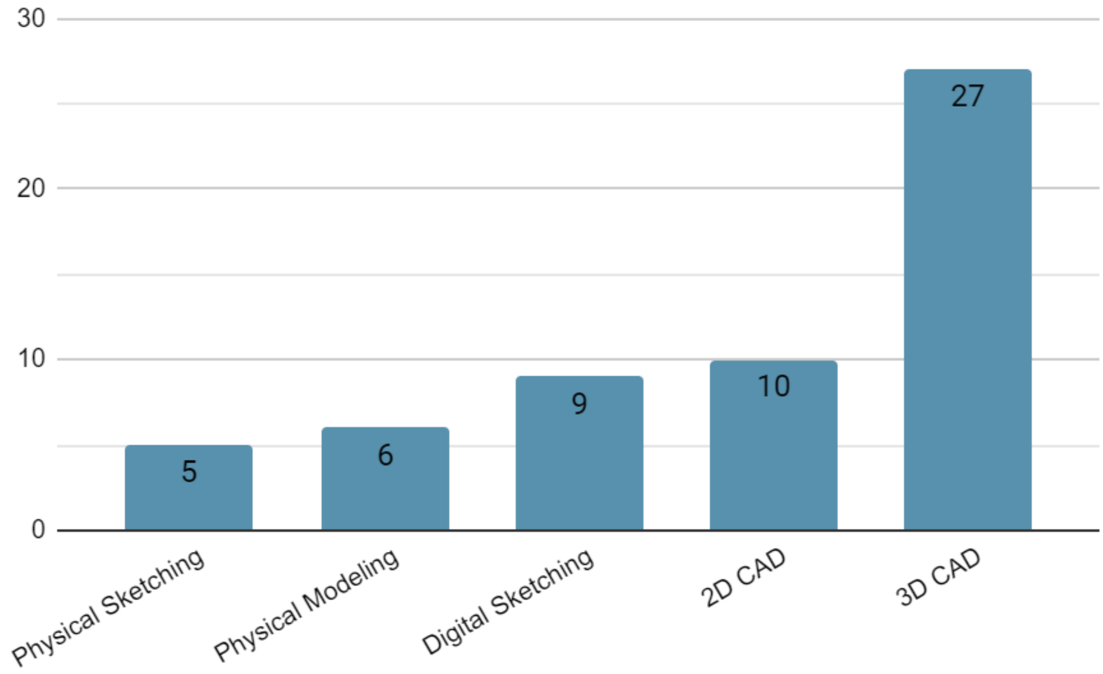

Figure 3.10. Answers for Question 10

Answers for question 10:

- Physical Sketching 5
- Physical Modeling 6
- Digital Sketching 8
- 2D CAD 10
- 3D CAD 26
- Other (2)
    - Rhino 1
    - Photoshop 1

Questions 11-15

The following questions were asked to determine the representation methods used in various stages of design. Similar to the previous group of questions, these questions were asked to find out if specific representation methods were favored over others in general or in certain stages of design. It was expected that the outcome of the responses would point to specific requirements.

Question 11:

Which of the following representation methods (showcasing) have you utilized?

This was a multiple-choice question, inquiring about all of the used representation methods.

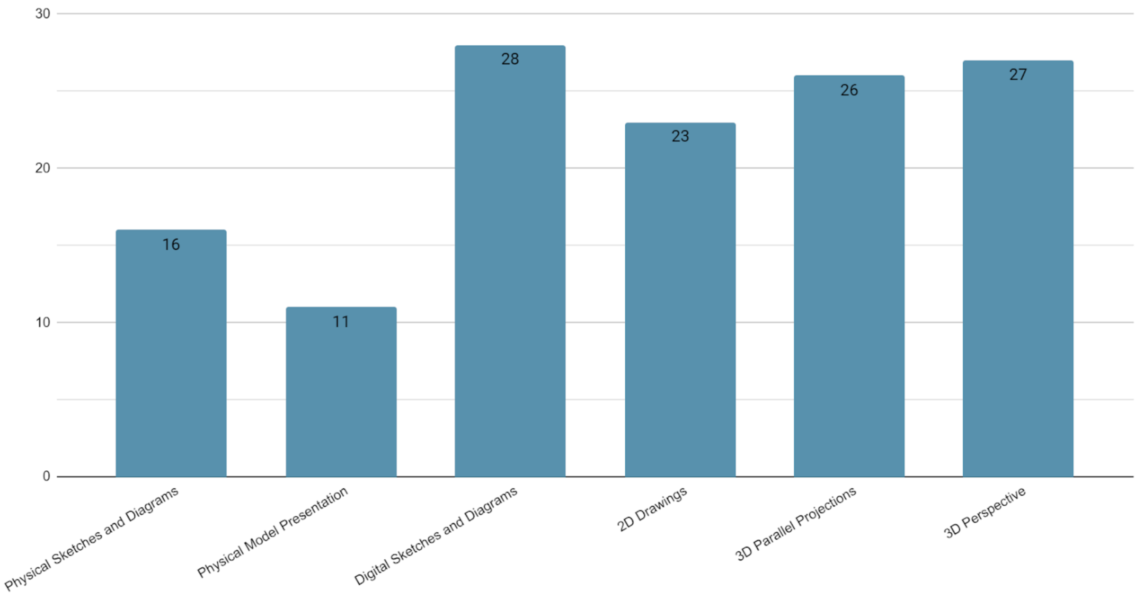

Figure 3.11. Answers for Question 11

Answers for question 11:

- Scanned Physical Sketches and Diagrams 16
- Physical Model Presentation (Photographs, Videos, Live) 11
- Digital Sketches and Diagrams 28
- 2D Drawings 23
- 3D Parallel Projections 26
- 3D Perspective 27
- Other (1)
    - Rendered Images 1

Question 12:
Which of the following representation methods (showcasing) have you utilized the <u>most</u>?

This was a single-choice question, inquiring about the most used representation method.

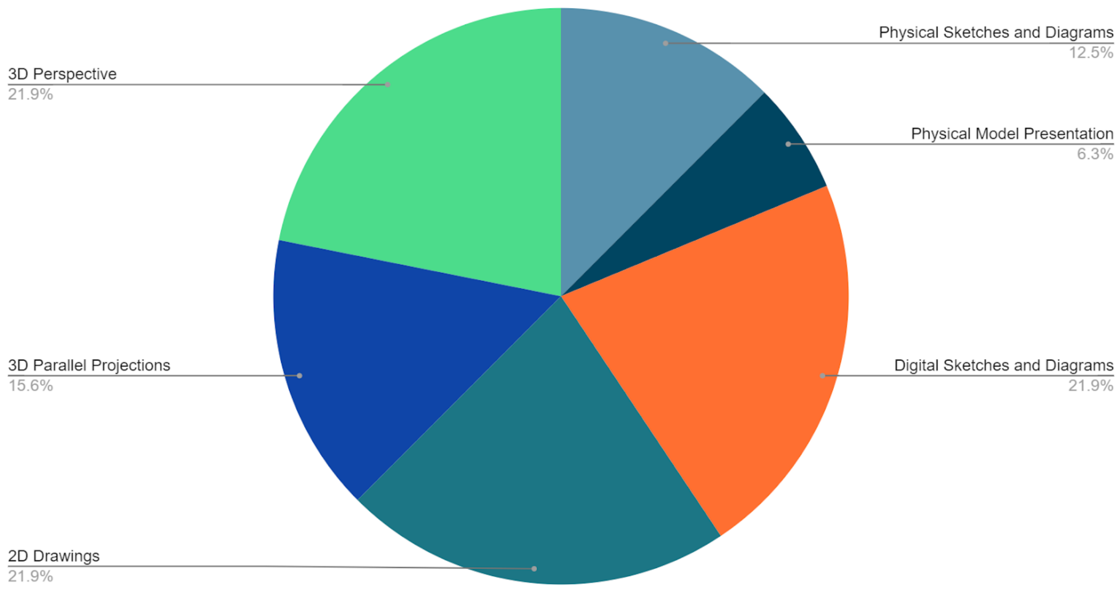

Figure 3.12. Answers for Question 12

Answers for question 12:

● Scanned Physical Sketches and Diagrams	4
● Physical Model Presentation (Photographs, Videos, Live)	2
● Digital Sketches and Diagrams	7
● 2D Drawings	7
● 3D Parallel Projections	5
● 3D Perspective	7
● Other	(1)
○ Rendered Images	1

Question 13:
Which of the following representation methods (showcasing) have you utilized <u>in the early stages</u> of the design?

This was a multiple-choice question, inquiring about all representation methods used in the early stages of design.

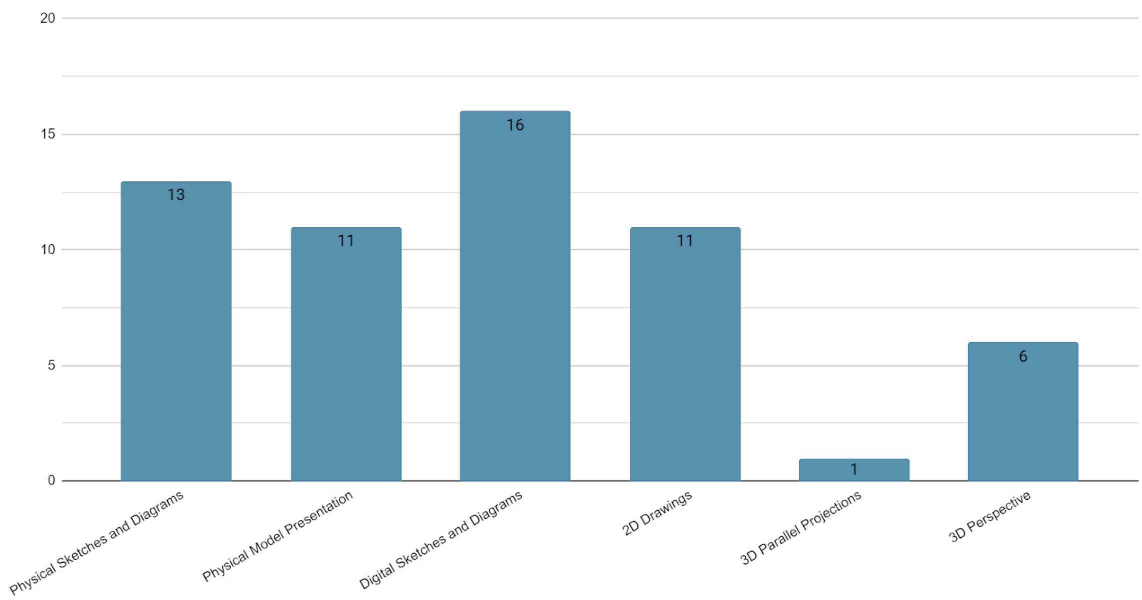

Figure 3.13. Answers for Question 13

Answers for question 13:

- Scanned Physical Sketches and Diagrams 13
- Physical Model Presentation (Photographs, Videos, Live) 11
- Digital Sketches and Diagrams 16
- 2D Drawings 11
- 3D Parallel Projections 1
- 3D Perspective 6
- Other 0

Question 14:
Which of the following representation methods (showcasing) have you utilized <u>in the final stages</u> of the design?

This was a multiple-choice question, inquiring about all representation methods used in the final stages of design.

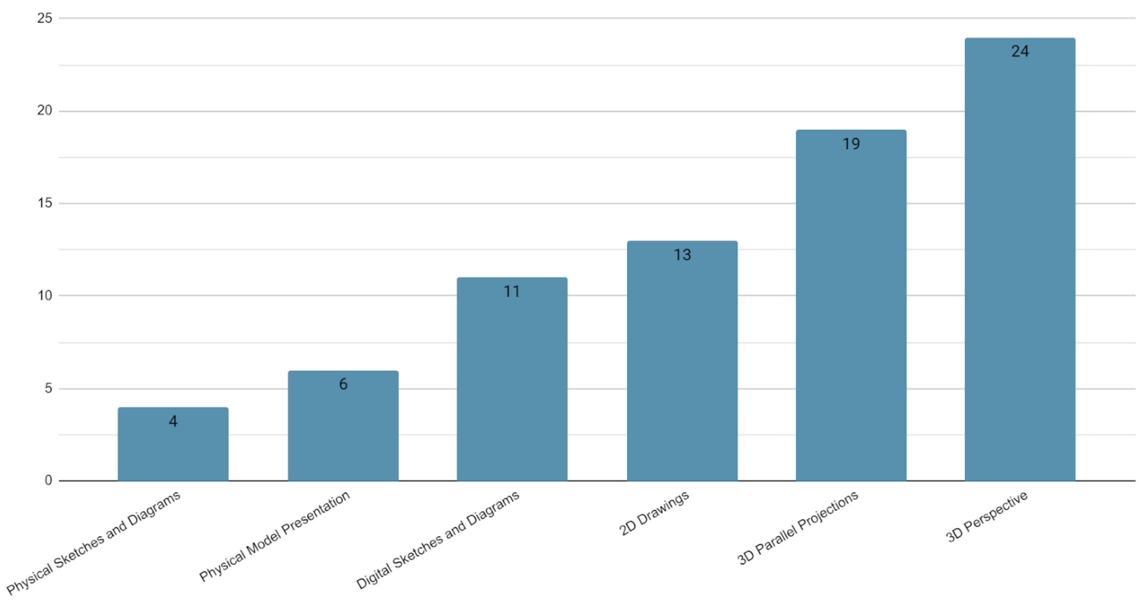

Figure 3.14. Answers for Question 14

Answers for question 14:

● Scanned Physical Sketches and Diagrams	4
● Physical Model Presentation (Photographs, Videos, Live)	6
● Digital Sketches and Diagrams	11
● 2D Drawings	13
● 3D Parallel Projections	19
● 3D Perspective	24
● Other	0

Question 15:
Which of the following representation methods (showcasing) was the most efficient to use while communicating in a remote learning environment?

The purpose of this question was to determine if certain showcasing methods are preferred over the others by the students in terms of communication efficiency. The outcome of this question was expected to provide a priority list in which implementation of certain features will be prioritized. This was a single-choice question.

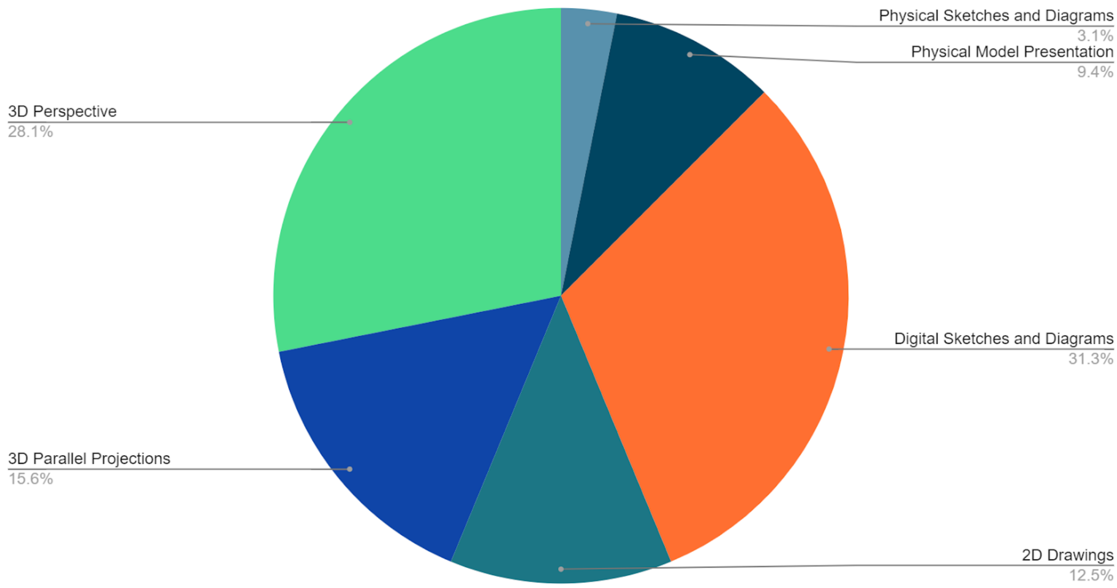

Figure 3.15. Answers for Question 15

Answers for question 15:

- Scanned Physical Sketches and Diagrams 1
- Physical Model Presentation (Photographs, Videos, Live) 3
- Digital Sketches and Diagrams 10
- 2D Drawings 4
- 3D Parallel Projections 5
- 3D Perspective 9
- Other (1)
    - Nothing 1

Question 16:
If you've utilized 3D representations, which of the following methods have you used while communicating with your studio instructor?

Assuming the research will be utilizing a 3D environment, this question was asked to determine if certain 3D representation techniques other than presenting their model in the 3D CAD environment were being utilized by the students. This was a multiple-choice question.

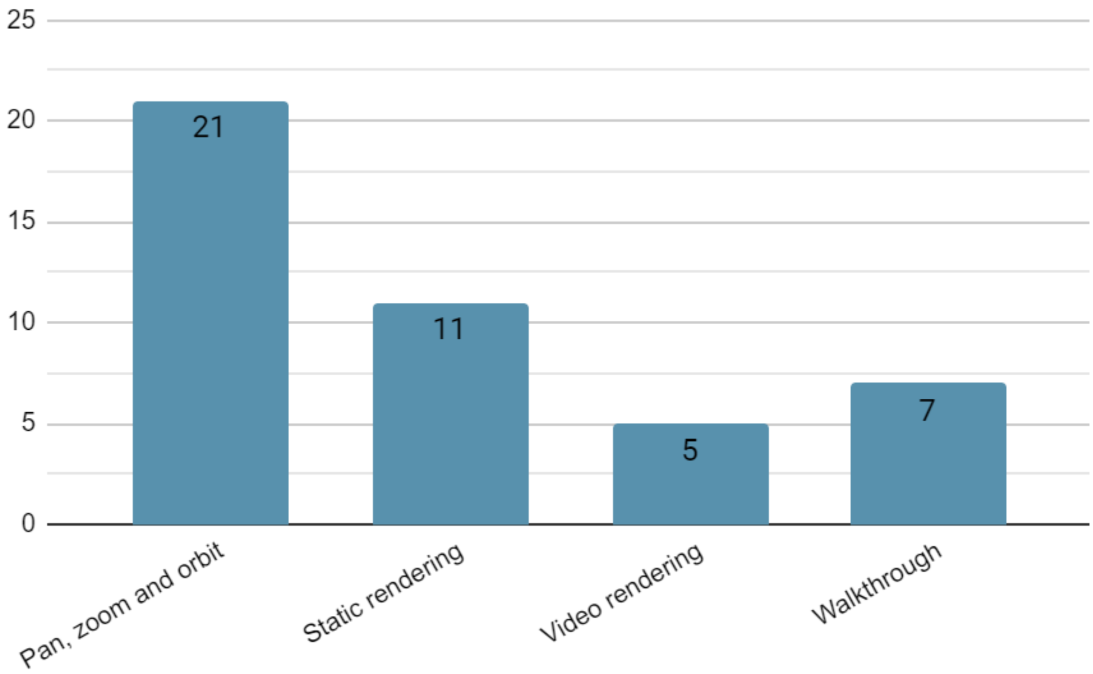

Figure 3.16. Answers for Question 16

Answers for question 16:

- Pan, zoom and orbit in the drawing environment 21
- Static rendering 11
- Video rendering 5
- Walkthrough 7
- Other (1)
    - Didn't use 1

The first choice, "Pan, zoom and orbit in the drawing environment", refers to presenting a model in the CAD environment itself. There are no additional steps taken to prepare a model for presentation, the model is viewed in the CAD environment where the drawing happens.

Static rendering refers to still images of the model from certain perspectives, with or without post-processing. Taking screenshots, using rendering software to produce images and similar methods can be included in this category.

Video rendering refers to moving images of the model in predefined paths, with or without post-processing. Rendering videos via rendering software or recording camera movement can be included in this category.

Walkthrough refers to the free-roaming presentation of a model. This technique differs from video rendering in terms of movement freedom. Observers can experience the model from a human perspective and walk freely in this method.

Question 17:
Were the studio sessions recorded by your instructor?

This question was asked in order to provide insight for the responses to the other questions. It was also asked to determine if session recording was common in the current distance education methods, which is usually not possible or preferred in conventional face-to-face studio sessions.

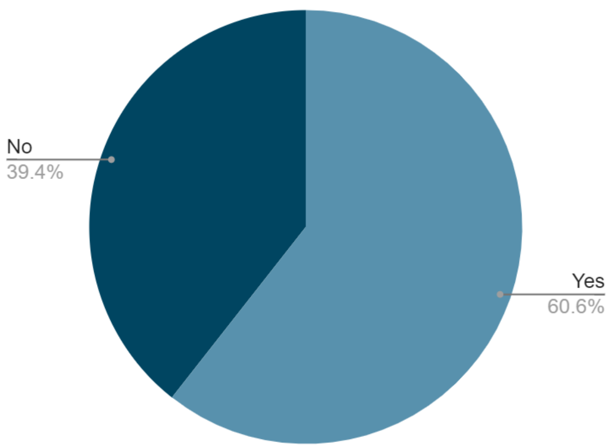

Figure 3.17. Answers for Question 17

Answers for question 17:

- Yes 20
- No 13

Insights for question 17:
Out of the people who answered "Yes" to this question, %85 of them answered "Yes" for question 21, which was "Have you been listening to the critiques given by your instructor to the other students?".

Out of the people who answered “No” to this question, %62 of them answered “Low” for question 22, which was “How would you rate the overall communication efficiency in the current remote learning methods?”.

Out of the people who answered “Yes” to this question, %70 of them answered “Yes” for Question 27, which was “Have you given feedback to or received feedback from other students?”.

Question 18:
If the studio sessions were recorded, have you been watching them regularly?

This question was to determine whether the recorded sessions were later watched by the students regularly. The adverb “regularly” is important to this question, the question itself isn’t inquiring about the frequency of the participants’ recording watching habits, but is rather questioning their own perception of the frequency of the act. The aim of the question is to determine if the participants really utilized this feature that is possible through the use of a software-based medium, no matter how relatively frequent. How they define the concept of regularity is left to the participants’ perception, it was assumed that if a participant perceives their watching habit as “regularly”, they were doing this act more often than occasionally, thus actually utilizing the provided capability. This was an optional, single-choice question. Although only 20 of the participants responded “Yes” to the previous question, 30 participants chose to answer this question.

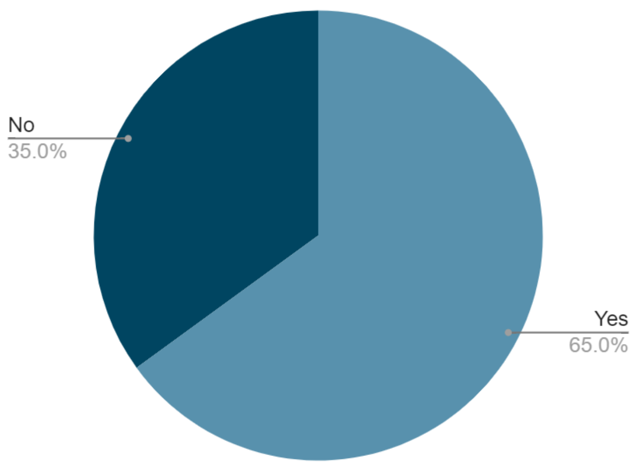

Figure 3.18. Answers for Question 18

Answers for question 18:
- Yes 13
- No 17 (7)

Given the number of responses versus the number of responses marked as “Yes” to the previous question, it may be assumed that the participants didn’t realize it was an optional question even though it was clearly marked as optional in the survey. Following this assumption and the previous question’s responses, it could be further assumed that out of the 20 responses for the previous question which state that their studio sessions were recorded, 13 participants watched the recordings regularly.

Question 19:
Which methods of communication were utilized by your studio instructor?

This question was asked to determine which methods of communication were more frequently utilized by the studio instructors in distance education. This was a multiple-choice question.

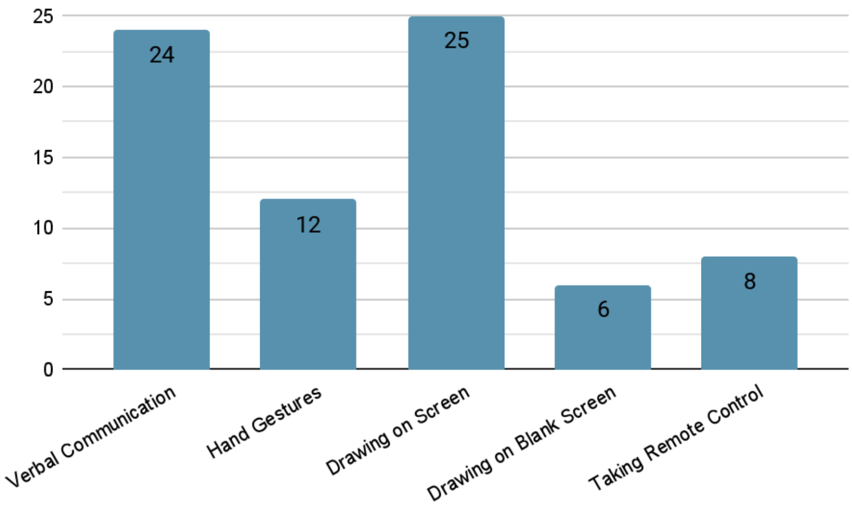

Figure 3.19. Answers for Question 19

Answers for question 19:

• Verbal Communication	24
• Hand Gestures	12
• Drawing on Projected Screen	25
• Drawing on Blank Screen	6
• Taking Remote Control	8
• Other	0

Verbal communication refers to voice communication over video conferencing software. In this type of communication, words are used to describe or discuss rather than utilizing other available methods.

Hand gestures refer to gesturing over video conferencing software. It was assumed that this method, which is also used in face-to-face education, could still be performed over the internet via video conferencing. Describing ideas with hand gestures with the help of a little imagination is not uncommon in conventional design studios.

Drawing on a projected screen is the act of using drawing tools provided by several video conferencing software, which lets users draw on shared screen video streams. This is considered to be one of the possibilities that distance education brought to design studios. Without the need of printing specific views or drawings, this method can be used to draw on top of the drawings or models in any given moment, in real time, visible to all observers and the presenter themselves. It is also possible to let multiple people draw at the same time.

Drawing on a blank screen is similar to drawing on a projected screen but this time not on the projected screen but on another blank screen that is usually provided by a whiteboard application within the video conferencing software.

Taking remote control is a method possible in some video conferencing software, which lets the users take control of other users' computers while they are sharing their screen. This method makes it possible to interact with the drawing or the presentation environment directly, similar to taking control of the mouse and keyboard in a face-to-face studio setting where the students are using their computers.

Question 20:
Which method of communication utilized by your studio instructor was <u>the most efficient</u>?

This question was asked to determine the effectiveness of the various communication methods utilized by studio instructors, in students' opinion. This was a single-choice question.

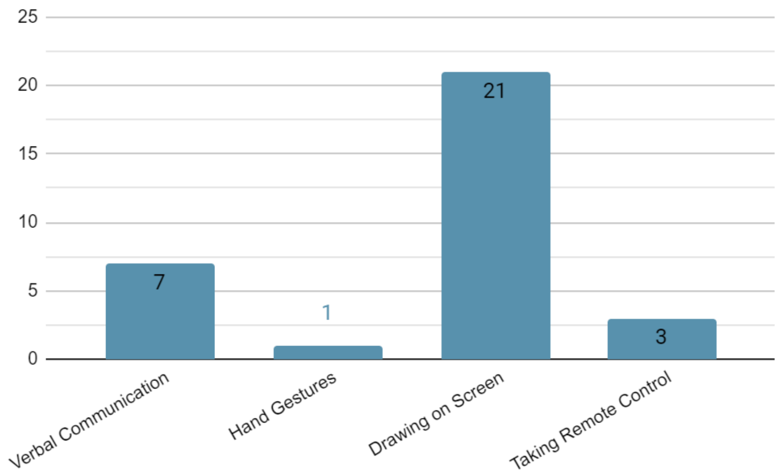

Figure 3.20. Answers for Question 20

Answers for question 20:

- Verbal Communication 7
- Hand Gestures 1
- Drawing on Projected Screen 21
- Drawing on Blank Screen 0
- Taking Remote Control 3
- Other (1)
    - "He wasn't efficient"

Question 21:
Have you been listening to the critiques given by your instructor to the other students?

This question was asked to determine if the participants were listening to other students' critiques as they would in a conventional face-to-face design studio setting, in the distance education model. It was assumed that the methods and the software used in distance education might be enabling students to skip such sections of design studio sessions.

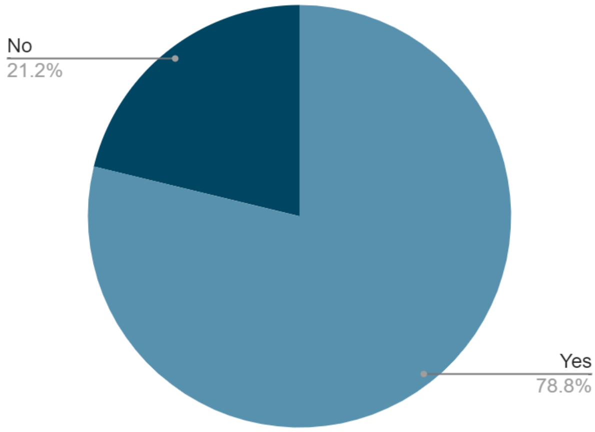

Figure 3.21. Answers for Question 21

Answers for question 21:

- Yes 26
- No 7

Question 22:
How would you rate the overall communication efficiency in the current remote learning methods?

This question was asked to determine the opinion of the students on communication efficiency in the current distance education methods. The outcome of this question was expected to point towards students' thoughts on the efficiency of the current methods. A low score would indicate that the students consider the current methods to be inefficient, thus showing that there may be problems in the current methods. This question was asked on a scale from 1 to 5.

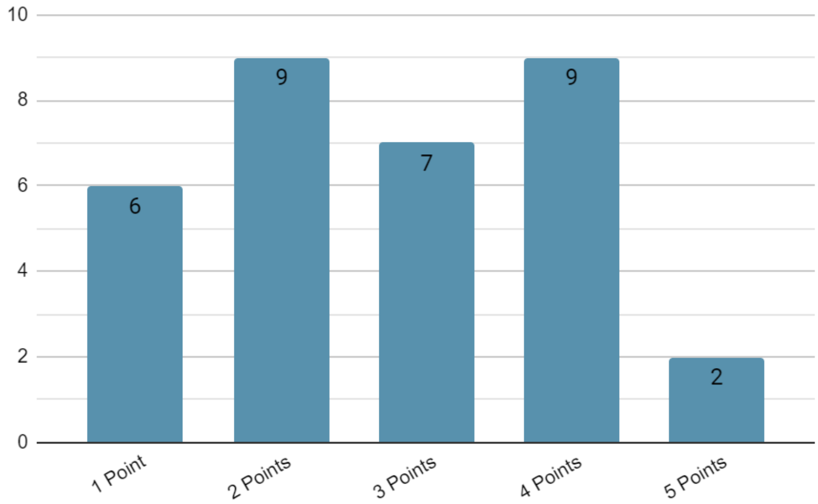

Figure 3.22. Answers for Question 22

Answers for question 22:
- 1 Point 6
- 2 Points 9
- 3 Points 7
- 4 Points 9
- 5 Points 2

Average Rating: 2.76

Question 23:
Have you considered user interaction in your design?

This question was asked to determine if participants take interactivity into consideration while designing. Doing so would require presentation techniques that could represent such interactivity. This would mean that the medium should also be capable of displaying these presentation techniques or offer methods to demonstrate interactivity. It was assumed that the majority of the students would be considering interactivity in their design, but wouldn't be utilizing proper demonstration techniques to represent such ideas, which is covered in the next question.

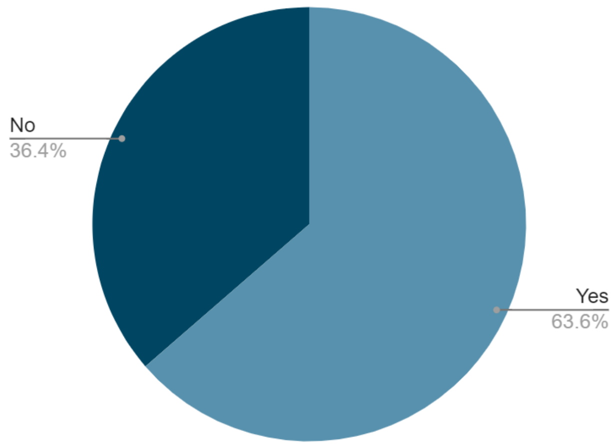

Figure 3.23. Answers for Question 23

Answers for question 23:

- Yes 21
- No 12

Question 24:
If you have considered user interaction in your design, how did you choose to represent it?

This question was asked to discover the presentation techniques that the students use in order to demonstrate interactivity. The outcome of this question was expected to point to the requirements of a medium to be used in distance education for displaying these techniques and if these techniques are up-to-date and appropriate. This was an optional question with a text box input.

Answers for question 24:

- Images, collages
- Images, sketches
- Gifs and diagrammatic sequences
- With storyboards (image sequences) that I took from rendered models
- Collages, montages
- Photo montages
- Powerpoint
- Image sequence, diagrams
- Images and illustrations
- Gifs, storyboards
- Videos
- Interactive models
- Diagrams
- Interactive models
- Interactive models
- Interactive presentation
- Videos
- I made a human model and made a stop motion video of it in my structure.

The answers were expected to be more detailed, similar to the last answer. It was assumed that the question wasn't properly worded to lead the participants into explaining their answers in detail. The answers didn't yield any useful information about the aspects of interactivity that the participants considered in their design, but the techniques for demonstrating such aspects were defined.

Question 25:
Have you considered the experiential aspects of your design? (e.g. time of the day, lighting, weather, acoustics, dynamic user circulation and population)

This aim of this question was to determine if the students were considering experiential aspects of certain conditions and situations in their design. Some examples were given in the question text to help differentiate between interaction, which was the context of the previous question, and experience. The purpose behind asking this question was to see the students' inclination towards using such aspects in their design.

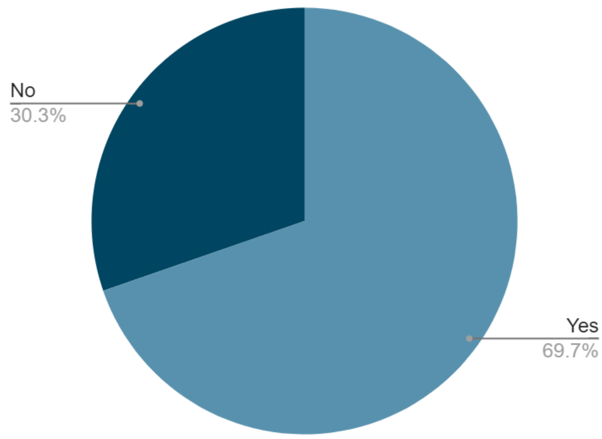

Figure 3.24. Answers for Question 25

Answers for question 25:
- Yes 23
- No 10

Insights for question 25:
Out of the people who answered "Yes" for this question, %87 of them answered "Yes" for question 21, which was "Have you been listening to the critiques given by your instructor to the other students?".

Question 26:
If you have considered the experiential aspects of your design, how did you choose to represent it?

As an addition to the previous question, this question is asked to determine the methods used by the students in demonstrating such aspects. The outcome of this question was expected to have implications on some requirements in terms of simulation methods. This was an optional question with a text box input.

Answers for question 26:

- Diagrams, sketches, digital models
- Sketches, images
- Image sequences, gifs
- Image sequences
- Image sequences
- Interactive models, videos, image sequences
- Photos or gifs
- Gifs, storyboards, diagrams
- Images
- Image sequences, videos
- Image sequences
- Image sequences
- Videos
- Image sequences, digital drawings
- Diagrams
- I took photos of it in different time periods and experimented with different internal and external light sources.

Similar to question 24, the responses weren't detailed enough to extract the aspects of experience the students had considered in their design. The answers were mostly about the demonstration techniques used to express such aspects. The answers to this question didn't provide any addition to the data collected via question 24.

Question 27:
Have you given feedback to or received feedback from other students?

This question was asked to discover if students have been giving each other critiques/feedback in the distance education period. It was assumed that the disconnection of the students between studio sessions would cause a decrease in this regard as the students no longer shared a common space besides video conference calls, which are only organized to last for the duration of the studio sessions.

There are also some technical aspects that were assumed to be causing further disconnection in this regard. Video conferences often support only one conversation to be held at an instance since all the users are connected in a single voice channel with no proximity filter. In a face-to-face design studio setting, multiple conversations can be held in the same space through proximity and voice volume. In a virtual video conference, all voices are heard and any attempt to hold separate conversations in the same room causes overlapping voices, thus incomprehensibility. It is possible to separate student groups into individual voice channels but that would cause disconnection from the tutors' side and further disconnection between the groups of students. This question was asked to provide insight on the severity of the problem, and was noted to be discussed further in detail later in the interviews.

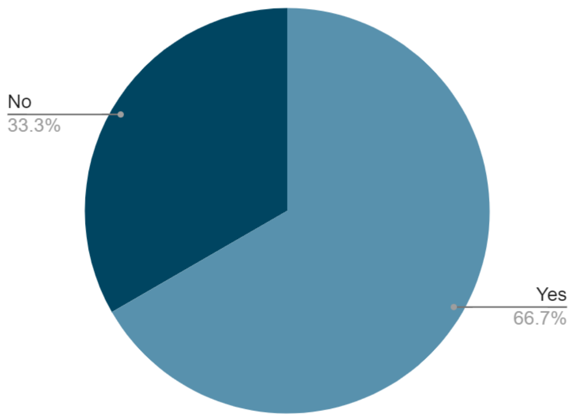

Figure 3.25. Answers for Question 27

Answers for question 27:
- Yes 22
- No 11

Insights for question 27:
Out of the people who answered "Yes" for this question, %96 of them answered "Yes" for question 21, which was "Have you been listening to the critiques given by your instructor to the other students?".

Question 28:
If you have worked in a group project, which of the following methods have you used?

This question was asked to determine the remote collaboration methods utilized by the students during the distance education period. This was an optional, multiple-choice question.

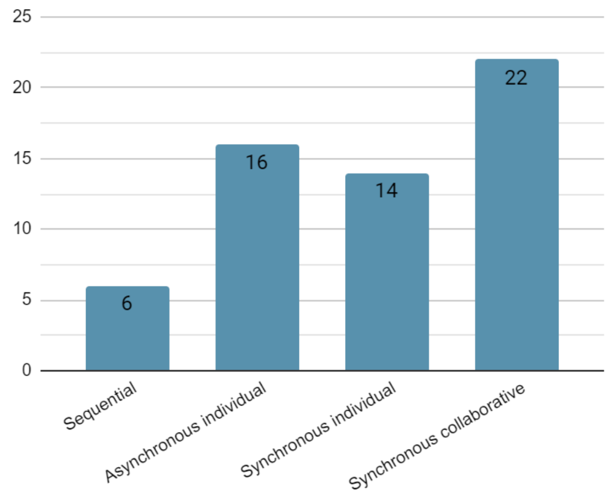

Figure 3.26. Answers for Question 28

Answers for question 28:

- Sequential working 6
- Asynchronous individual working 16
- Synchronous individual working 14
- Synchronous collaborative working 22
- Other 0

Explanations to the methods were given to the participants inside the question text.

Sequential working refers to working on a single file, one person at a time, sequentially.

Asynchronous individual working refers to working on individual aspects of a project, then merging work, without real-time communication.
Synchronous individual working refers to working on individual aspects of a project while communicating, then merging work.

Synchronous collaborative working refers to working collaboratively in real-time, on the same file/document whether through screen sharing or the utilization of a software capable of real-time collaboration.

Question 29:
Which analysis and simulation methods have you used in your project?

This question was asked to find out if students have been using any analysis and/or simulation methods in their projects. The outcome of the responses was expected to point out to a priority list when defining requirements in terms of simulation and analysis possibilities. This was an optional question with a text box input.

Answers for question 29:
- Grasshopper sun path analyses, wind circulation
- Weather consultant for weather data
- Rhinoceros rendering

The lack of responses and misunderstanding of the question yielded no meaningful result for this question.

The next two questions were optional, text box input questions. They were left open-ended to let the participants comment however they wanted. The aim of these questions was to unveil the students' opinions on the advantages and disadvantages of distance education.

Question 30:
What were the advantages of the current remote learning environment?

Answers for question 30:
- "Working more efficiently but I can't say the same for listening."

This answer seems to point towards concentration issues when working from home. It is assumed that the respondent is feeling that they can do their work more efficiently but can not concentrate while listening to lectures.

- "Not dealing with problems caused by models on the roads."

This answer seems to address the transportation issues faced while moving physical models.

- "It had no advantages, unfortunately."

- "Communication with teachers got better since we were all in our homes, so it became a bit more informal. Also I have learned to decide on a working schedule myself."

This answer seems to address behavioral differences that occur when communicating from home. They state that the conversations become more informal. They also declare that they have learned how to schedule themselves while working from home.

- “I think I have learned a lot in computer programs such as Photoshop, Rhino and Illustrator because of the digital environment.”

- “It is easy to access. It is relatively cheap. It doesn’t rely on mobility.”

- “I learned a lot about digital aspects of design.”

- “Escaping from the traffic and using that spare time to further develop design.”

- “Saving time on transportation, not having to print posters.”

- “Only advantage is taking extra courses.”

- “It’s good that we can listen to the lectures again.”

This answer seems to point out the fact that distance education has increased the inclination towards recording lectures and that these recorded lectures can be revisited by doing so, which was found as a positive outcome of the distance education period by the student.

- “Learning to adapt to multiple methods. Realizing which professors and students put in academic effort. A dream space for introverts.”

In the last sentence, this response seems to address the decrease in social anxiety when working from home and the possibility of taking education without being in the physical proximity of other students.

- “Individual focus. Pushing the limits of online learning to succeed more. Using more online tools. Versatile documentation in projects. 7/24 communication with tutors. Also no socializing and just studying on the project might get hard at the beginning but that’s what makes the project better in the end.”

In their last point, the participant seems to indicate that the lack of communication between students in distance education might decrease their ability to socialize but increase their projects’ quality.

Question 31:
What were the disadvantages of the current remote learning environment?

Answers for question 31:

- "Interactive learning or practicing the realm is pretty hard. Staying just at the theoretical phase is not enough. Practical working is needed and would be more efficient. In this case, I would recommend slowing down the syllabus or giving one more year for architecture departments. 4 years is not enough to learn practical and theoretical ways even in face-to-face education. So having remote learning is harder and isn't efficient."

This answer seems to indicate that distance education has eliminated the ability to have hands-on experience in design education. It may be safe to assume that the respondent is most likely from the industrial design department as their design studios rely heavily on physical modeling and fabrication.

- "Not having 100% concentration."

- "Since we could not get the flow of information, I could not reflect this to the projects. Also, our motivation was low. I did not participate in the class and neither did my friends. There are a lot of students who couldn't use the digital platform and thus couldn't learn."

This response raises two important issues. Firstly, they imply that the students' motivations were lower than usual which caused them to participate less in the classes. Secondly, they indicate that some students weren't able to use the distance education tools.

- "Lack of focus on lectures due to online medium."

- "The studio is not only a place where we do individual work, it is a space that we have shared with others, especially with students from different levels. There was a lot of idea sharing happening, we have lost the "peer communication". Also it is hard for us to present our projects and it's hard for our teachers to comment on them since they can't sketch easily."

This response points out to the fact that students have lost their ability to communicate in distance education, which resulted in the inability to share ideas and critique each other. The respondent also indicates that giving critiques has been harder for tutors since they cannot just draw on printed sheets as they would in a face-to-face setting.

- "Group projects are done less, student interaction is less in class and out of class. Explaining physical models is a little harder."

- “Not as efficient as face-to-face education and causes lack of motivation.”

- “Lack of studio environment. Lack of bigger group projects. Less interaction with other studio sections. Smaller and shorter juries.”

- “Not being able to do physical models and not being able to use workshops. Communication and internet problems, and my attention span is 4 seconds maximum now.

- “Face-to-face was sometimes more effective in terms of seeing other students’ progress.”

- “Not being able to communicate with the tutors and classmates. Not being able to focus and follow the course properly. Not being able to socialize and share ideas.”

- “Not efficient.”

- “Not being able to see everything at once.”

This response seems to indicate the technical issue of the inability of having their own perspective in the distance education method they have used. Observers have the ability to observe anything in the studio space in a conventional physical studio environment, whereas in most applications used in distance education, that ability doesn’t exist and all the observers are forced to see through the presenters’ view.

- “Inability to speak with tutors privately and face-to-face regarding certain issues.”

- “The communication between tutors and students was way too difficult.”

- “I cannot focus for long periods. I cannot discipline myself to work without external factors.”

- “No socializing, being unable to learn from each other easily. Having no physical access to each other's work. Also instructors have got no physical access to our work. Not enough peer motivation and interaction. Zoom fatigue. Mental exhaustion.”

3.2. Design Studio Tutor Interview

Following the student survey, a semi-structured interview was conducted with 6 tutors who had experience in teaching during the distance education period in all departments and levels of the Istanbul Bilgi University's Faculty of Architecture. The interview lasted 83 minutes and was conducted on 29/06/21, which is after the first year that was completely conducted according to current distance education methods.

Without explaining the intent of the study, the tutors were first asked about their experiences during this period, in terms of the difficulties they have endured and the advantages they have felt. Follow-up questions were chosen according to the outcome of the previous questions, either to further elaborate the statements or to surface the underlying issues.

Starting from the experience with freshmen of the Faculty of Architecture, who attend the studios "Basic Design I" and "Basic Design II" collectively, it was stated that in this level, the studio is more focused on physical production of models in parallel to being acquainted with model-making materials. It was stated that because these first-year studios conventionally formed around this approach, distance education methods couldn't reproduce the studio in the digital environment. Instead, students were asked to practice model-making with the materials that they could easily find around them. It was commented that a shift from physical to digital occurred naturally, though this occurrence impacted the students' introduction to the design studio as a physical space, in which students worked together while communicating. On another issue of impacted introductions, it was stated that the students missed their chance to learn about fabrication tools such as CNC routers, laser cutters and 3D printers which are present in the fabrication laboratories of Istanbul Bilgi University and frequently used by the students.

As expected, and without further direction, the issue of scale perception was pointed out immediately after the previous comments. This was one of the main predictions on the issues that would arise in a digital design studio, that would be asked if not naturally identified in conversation. It was stated that conversing through a digital interface impacted their ability to communicate on scale.

When asked if they had perceived any advantages in the distance education methods, it was again stated that the interaction between students in the physical design studio was lost due to the circumstances, and that meeting inside this physical design studio was absolutely necessary for freshmen to be introduced to design studios which persist throughout the architectural education. It was explained that in an attempt to make up for this interaction, the students were asked to group up and work together if possible. This yielded an unexpected result where it was common to have large groups working together in the conventional Basic Design studios, the small groups which were formed during these terms outperformed the larger groups of the previous face-to-face terms.

On the other hand, concerning the senior students of the department of architecture, it was stated that some issues in the scale perception was also common among the student projects. However, these issues were perceived as advantageous as well as disadvantageous. Advantageous in the sense that the students tend to design much larger projects in programmatic scale than they used to, and disadvantageous in the sense of conventional scale perception, as in the projects were unnaturally large in spatial qualities. It was also stated that for the first time, megastructures were designed in this period, among senior year students. This was influenced by some studio tutors as they decided that since they won't be physically doing field trips to a site, they would rather use this in advantage to have students propose utopian megastructures regardless of a physical site. On the other hand, another tutor stated that the inability to go on these field trips was the main cause of the problems in scale perception. It was stated that while these trips were crucial in understanding the site, they were also critical in perceiving the scale of it. However, being unable to physically visit a site was found influential in increased GIS software usage among students. It was stated that, rather than collecting aerial images and using the drawings provided by their tutors, students showed an interest in using various software to better understand the sites.

On the issues about communication and presence, students being present on a video conference session simultaneously was found advantageous, although the complete lack of the conventional design studio environment was thought to have entirely eliminated the interaction between students, which was found more important than the advantages of presence. The lack of interaction between students was thought to have forced the students into working individually, which is the opposite of how it used to be, the students would work in groups throughout their entire education, even in their thesis. It was explained and discussed that a physical, common studio space is crucial in having students communicate with each other on their projects, learning from each other and having secondary discussions between themselves, rather than solely receiving critiques from their tutors. One tutor explicitly stated that they think that the conversations between students in such an environment is more important than the ones between the students and themselves. It was also discussed that the students who kept their cameras on would naturally be more present during conversations and were more active during classes.

It was later stated that students being present on a video conference session didn't mean that they were always there. The students would often listen to their part of the critiques and disengage from conversation, perhaps not even being physically present in front of their computers.

An impact in the communication quality was discussed, being unable to use gestures, or pointing and discussing in a shared space was used as an example to indicate such a problem. It was stated that both the students and tutors became efficient in using the provided drawing tools in video conferencing software though it wasn't found as efficient as the conventional pen-and-paper approach or the aforementioned methods.

On all levels, it was discussed that physical model-making was severely impacted. The inability to pick up, rotate and alter the physical models on the other side of the screen was also thought to have impaired the quality of communication through these models. Inability to produce scaled physical models was also discussed to have damaged the perception of scale among students, which was also tied to projects becoming larger than usual. This was found to be most problematic among second year architecture students, which is explained to be significant in transition to architectural design after the first year studios which focus mostly on basic design principles. On the other hand, tutors felt that an advantage that arose from this disadvantage, was the reallocation of the time spent making these models into other areas such as building a theoretical background for the projects.

A positive impact on design communication was found by the tutors in the distance education model. As the students and tutors were no longer present in the same space and with their working material at hand, the students were influenced to prepare presentations for their material more often. This was tied to the students being unable to have spontaneous conversations on their working material, which is a common way of communicating between tutors and students that would happen with on-the-fly sketches, gestures and abstract definitions of what would or could be. It was thought that the lack of such communication methods pressured the students to have proper presentations of their work whenever they needed to present and discuss their work.

A loss of concentration among students was also discussed. It was stated that one of the reasons for this could be the lack of the feeling of being a part of a community through being isolated in their rooms during this period. The lack of surrounding students, feeling alone while working was thought to have discouraged the students from working. It was further explained that the design studio environment produces an isolated space for students, the example of a movie theater was given to explain this isolation and concentration through isolation, through similarity.

It was explained that in every studio, a certain portion of students would stay in a comfort zone defined by critiques given by their tutors. It was stated that the percentage of these students increased in this period, thus reducing the amount of "risk takers".

As an interesting comment, it was stated that through being only engaged with the university in class hours, the university became "too serious". It was explained that the students lost their ability to socialize and have fun with each other between and during classes. This was also tied to the loss of communication between students. It was stated that this caused the informal conversations between them and their tutors, which could improve them in numerous ways, to be completely lost.

Regarding industrial design studios, second year students were the most impacted due to distance education. It was explained that in the second year studios, students would work physically in various craft workshops as apprentices. In an attempt to simulate this process, the students set up their own workshops inside their homes but it wasn't found to be sufficiently educational as the tools that they could acquire and use were limited regarding costs and safety. It was stated by the tutor that especially for industrial design studios, a digital design studio isn't a viable option as their education depends on physical approaches.

On a more technical side of the interview, it was stated that having to follow multiple windows such as a group drawing board, participant cameras and presentation material at the same time was found to be too distracting and difficult.

Similar to the issues about scale perception, it was stated that the presentation material and poster designs lost the sense of scale and proper layouting. It was further explained that while presenting their work, students would scroll through their material at their own pace and it would become hard to follow in any other pace due to the lack of individual perspective, which one would have in a physical presentation environment. This issue was tied to the lack of proper layouting rather than the inability to observe through individual perspective, by the tutors. When asked if they had tried doing the presentations on groupboard applications where users have their individually controlled perspectives, it was stated that they have utilized both methods. When further inquired about the issues on presentation scales and layouting, it was revealed that rather than designing their presentations and posters with digital media and distance education methods in mind, the students were forced by the tutors to stay loyal to paper sizes and scale their material accordingly, as they would in a conventional face-to-face studio where they hang physical posters. It was later explained that by forcing students to prepare their presentations in a certain size, it is expected that the students would lay their work in a composition with an order of narrative and importance through size and positioning on this fixed-sized canvas. Aside from the scenarios where the posters actually have to be plotted.

As a side note, having the opportunity to easily have participants, such as guests and other lecturers without time and distance constraints, connected to class sessions was found to be one of the advantages of distance education.

3.3. Student Interview

Following the design studio tutors' interview, the tutors were asked to randomly select students from each of their studios to participate in a student interview. The students were contacted and invited to participate in the interview, out of the 27 students contacted, 6 were able to participate. One of the students was an architecture freshman, who had their whole university education until that day via distance education.

The student interview was done on a semi-structured basis, similar to the previous interview, with hidden references from the previous tutor interview and the student survey. The interview was conducted on 07/07/21 and lasted 161 minutes.

Students were first asked questions parallel to the survey, starting with the software they preferred. It was stated that every one of them were comfortable with using Rhino in classes, adding that it was expected of them since they were taught Rhino in Basic Design I and II. A surprising revelation was made at this early point that they also used software such as Revit in their workplaces and that they hadn't taken their habits while professionally working into consideration while answering the survey. They also pointed out that they do not prefer working with Revit in education as the workload becomes unbearable when working with such software as the design studios revolve around iterating design options and BIM software are not suited for such rapid prototyping.

Software such as Blender were brought into discussion, when inquired about to what extent they used Blender, it was revealed that they preferred to use it for simple 3D modeling and rendering. While Blender is strong in 3D sculpting, subdivision modeling, rigging and animating, none of the Blender users utilized these opportunities.

An industrial design student pointed out that Rhino wasn't ideal for them and that they would prefer to have learned software such as Solidworks rather than Rhino. This was tied to the fact that they usually design smaller objects than architecture students and they usually include moving parts or mechanisms in their projects.

When inquired about control schemes, students stated that they were more comfortable with pan and orbit controls while both drawing and presenting. It was stated that because it became harder to communicate on a digital model in distance education, walkthroughs and associated software and control schemes were not preferred. They explained that because of lagging video streams and the difficulty in explaining desired position and rotations, walkthroughs and inside-the-model views were harder to communicate on. It was also stated that this influenced the students to produce static views and renderings rather than directly viewing a real-time model in a CAD or rendering software environment.

Industrial design students stated that because their design studios are focused on physical works and working in real-world craft workshops, they rarely use digitally produced media for communicating in design studios. They would often use digital modeling tools only to plan their design and later fabricate and communicate on the physical prototype directly. However when they do use digital media, they stated that they prefer to have photo-realistic renderings to accurately represent their designs, they explained that they pay attention to details such as light refraction and wood grains, which they found is only viable to simulate within photo-realistic renderings.

In contrast to not having a single mention of it in the survey, an industrial design student stated that they also use advanced simulation techniques such as particle physics to simulate the airflow through an object that produces certain smells.

When inquired about utilizing camera projection techniques and adjusting camera properties, all students were aware of the difference between axonometric projection and perspective cameras but only the senior year architecture students had tinkered about camera properties such as focal length.

On the subject of presence and concentration in distance education, students immediately stated that they could not concentrate when they weren't required to have their cameras on. It was explicitly stated that having their camera on was the only way for them to feel present in the class and stay concentrated.

An important point was made by one of the senior architecture students while talking about concentration issues, they stated that because the tutors tend to continuously teach and talk on a subject in online classes, informal conversations were rarely held and without these small breaks in the classes, students would often lose concentration after some time from being overwhelmed. Due to the inability to chat with their classmates or initiate off-topic conversations with the tutors in these sessions as they would in face-to-face classes, they stated that they usually do house chores or surf on the web when they feel overwhelmed.

When asked about their thoughts on camera obligations, they stated that such obligations help with their concentration issues and keep them present in the classes and they would enforce it but they wouldn't prefer it to be an obligation as students. They also stated that when it is not an obligation, they tend to turn their cameras off after some time and lose their concentration, inevitably.

From an emphatic point of view, a student suggested that keeping students' cameras on was the only way for the tutors to feel students presence, they stated that they have had experiences in which tutors felt that were teaching to nobody as all the students' cameras were turned off and asked the students to turn their cameras on to see that they are present. After this statement, more students joined in on this comment and stated that they felt that their tutors' motivation and the quality of the lectures also fell when they didn't feel the students' presence.

When asked about their habits and methods used while designing and presenting in various stages of their design, they stated that even when their environment became more digital, their preferred methods of drawing became more analog. In one example, a student explained that they used to digitally sketch on a tablet before distance education, but as their environment became more digital, they perhaps subconsciously started to prefer to do hand sketches on paper in an attempt to acquire some feeling of balance between analog and digital.

Moving on to how the involuntary distance education period has impacted their education, a freshman architecture student stated that they have felt that they missed out the opportunity to learn how to operate and use digital fabrication tools such as laser cutters and 3D printers. It is worth reminding that in Istanbul Bilgi University, the first-year studios are focused on getting acquainted with various model-making materials and using such fabrication techniques. However, senior architecture students felt that there hasn't been much impact on their education regarding the commodities offered by the university as they were already more inclined towards digital drawings and presentations at this level.

When asked about how they managed to present physical work such as physical models and hand-drawn sketches in a digital environment, students stated that they took videos and pictures of them and prepared digital presentations based on these images. They added to this by commenting that this period had taught them how to take proper photographs and videos by requiring them to take this approach. It was also discussed that by using better angles, lighting and proper photography techniques, less successful designs were presented in a way that helped them score higher grades, and vice versa. This was perceived as both an advantage and a disadvantage.

On the subject of material qualities, students found it hard to explain how a material felt or acted in certain situations through the medium. In one example, an industrial design student stated that one of their models was made out of a sugar-based material and that they had made a candle holder out of this material. When asked about how it reacted to heat and the fire from a candle, tutors found it hard to believe that it wouldn’t catch fire or melt but the student explained that it wouldn’t be a problem since it wasn’t being heated from the candle due to its position. They stated that they couldn’t find a proper way to express the travel of heat through the medium, where they could just feel it with their touch if they were face-to-face, which left the tutors in disbelief.

After talking about physical models and materials, the subject of scale was brought up by the students. A senior architecture student stated that their scale perception is tied to these physical models until third year. They added by saying that the transition to digital from physical in the conventional studio environment also brought some problems in scale perception. Cutting shapes and forming compositions according to a scale would keep their scale reality in check but when they transitioned to a completely digital modeling environment, they often misinterpreted the sizes they saw on screen. They explained that they found physical model-making in the first years to be necessary to help with scale perception. However, as their projects became larger in the following years, physical models would become harder to produce due to smaller pieces with smaller scales, material and fabrication costs and time consumption, thus they would become undesirable. They stated that they found physical model-making obsolete at this point and without physical models, the problems in scale could only be solved through practice and self-checking.

When asked about how they preferred to present their work in a digital environment, all students stated that they preferred to prepare video presentations of their material. Industrial design students backed this statement with the necessity to explain moving mechanisms and displaying the effects of a dynamic environment, and architecture students stated that they felt that this technique would better explain their projects with more points of view rather than being bound to a few angles in static images. Although architecture students feel that video presentations are the most efficient form of presenting their projects, they also state that it is usually preferred for their more important presentations such as midterm and final juries due to the fact that it takes a large amount of time to produce and edit such videos. Having four sides of a model in an axonometric view was also found efficient among architecture students, considering the time and effort required.

On the subject of recorded sessions and if these recordings were watched by the students, most of the students stated that if the sessions were recorded, they tend to watch them afterwards and they preferred the sessions to be recorded. When asked about the reasons behind them watching these recordings, they explained that they would often miss some points while listening and watching the recordings would help them better understand what was being discussed. Inquiring about whether they only watch the parts concerning them or the whole recording, students stated that they often watch only the parts associated with their own work or a general discussion held in the studio. Some students however, suggested that having the sessions recorded decreased participation in online classes and should not be done, this was explained by that having recordings often lead students to not follow the sessions in real-time. They added by stating that in a face-to-face environment, students are obliged to follow the class attentively and that it is in their responsibility to do so.

When further asked about whether they listen to critiques given to other students, but inside the live session, students were divided in their opinions on this subject. Some suggested that listening to other discussions was as important as listening to theirs and would stay for the whole duration of the class, and the others felt like it was a waste of time and would rather not stay for that duration if possible. For the students that don't like to listen to the other discussions, whether they stood or not mostly depended if there was a predetermined order for discussing projects. If there was, these students would tend to only participate when their turn was nearing and would be absent when they know that they wouldn't be missed or asked about.

On a relevant subject, students were asked if they had given critiques to each other while hinting that the students ability to interact with each other was impacted in this period, based on the tutors' comments. Freshmen architecture students commented that they had felt this impact, declaring that they didn't have the chance to inspect other students' works outside of studio hours unless they were friends. They also explained that most of the freshmen worked individually rather than in groups, opposed to students usually working in groups in the previous, face-to-face terms. This was explained by the tutors in the previous interview to be a tradition among architecture students, where they would prefer to work in groups until graduation. Senior students stated that in the previous terms, some outliers among their friends would work hard but not be present in the discussions held in the studio. They continued by adding that these students cut their ties with the studio completely in the distance education period and performed much worse than they used to. They commented that what they think these students cling onto was the "vibe" of the design studio, even if they seemed detached from the environment; which wasn't present in this period due to the lack of a common space between students. The senior students further commented that the ones who were actively present in the studio managed to continue to do so in the distance education period, in giving and receiving criticism from other students.

On an interesting note about watching recordings and inspecting other students' works, some students stated that they do not prefer their work to be easily and unlimitedly observable by other students as it sometimes tends to influence other students to copy their work when it was praised by their tutors. This was discussed explicitly on viewing session recordings and being able to see shared files on the cloud where students were obliged to do as classwork submissions.

Apart from architecture students' experiences, the participating industrial design students stated that in their studio groups, they had video conference sessions between themselves where they would help each other out by discussing their projects, giving and receiving feedback, and interpreting the tutors' comments while reminding them. This was found very constructive by these students and was thought to boost the quality of their projects while also being enjoyable.

On the topic of communication efficiency, the participating students stated that they found on-screen annotations to be most efficient in student-tutor communication. One student stated that they found it hard to follow such annotations and preferred written explanations through emails. This was tied to the on-screen annotations being temporary and not accessible except through session recordings. They explicitly stated that they preferred permanent forms of comments and annotations. A further comment was made regarding this statement, another student who had the same experience explained that written comments might be more efficient due to the fact that they are not spontaneous as they are in a live conversation, but rather well thought out due to being written in a spare time. They also stated that in such emails, the content is also richer with elaborate explanations, examples and drawings due to the same reasons.

As a side note on this subject, students felt that when they had uploaded their work to the cloud prior to presenting, tutors had the time to carefully examine their work and better comment on them.

Moving on to another subject, the students were inquired about the methods they utilized to work on group projects. Starting by stating that tutor-decided groups were extremely unpleasant and inefficient to work with especially during this period, they explained that this was mostly caused by the inability to feel pressured by their peers or a sense of responsibility when not being present in the same physical space. They have given examples on how their group mates could easily go missing for hours or even days, or spend minimum amounts of effort without peer supervision. This was tied to the disconnection from only being connected through digital interfaces where one can easily hide and isolate themselves from all responsibilities, without having to face any of their group mates and related consequences.

On the contrary, students who had the opportunity to form their own groups felt no problems in working with groups during this period and found it rather enjoyable, educational and efficient.

On the subject of the methods utilized while working in a group, the students who worked with tutor-decided groups found it most efficient to work on separate parts of a project and then merge them, as in asynchronous individual working. The students who formed their own groups had video conference sessions and worked collaboratively as much as possible through sharing screens and annotating. When inquired if they would prefer a platform where they could collaboratively work in real-time, such as a 3D modeling environment where they could work simultaneously, they stated that they would very much prefer that and it would almost feel like working collaboratively in a face-to-face environment.

When asked about representing material qualities, experiential aspects and the user interaction possibilities in their projects, industrial design students stated that they found it hard to do such things in a digital environment and they would usually physically demonstrate such properties in the conventional studio environment. They added by explaining that they tried to film such interactions but it required too much effort than usual which they felt wasn't worth the effort. Architecture students, on the other hand, stated that they were already used to creating image sequences of such aspects and that they felt no difficulty in doing so in this period.

Referring to some of the issues pointed out by the tutors in the previous interview, the problems regarding scale perception were brought up again to further elaborate on the issue through questioning students. Students were inquired if they felt any problems in their scale perception or if they had any errors concerning scale in their recent projects. Industrial design students stated that since they work on smaller objects, they haven't had any issues on the matter. Architecture students however, described that not being able to have field trips and see the actual scale of a site themselves have caused problems regarding this issue. Freshmen architecture students have stated that they made errors in scaling and that it was mostly due to not being able to fabricate any 1-to-1 scale prototypes to perceive the actual size of their design, which they used to do using the fabrication facilities inside the campus prior to this period. During this conversation, a higher-level student asked a freshman if they had done an exercise where they would model their room and compare their design to the size of the objects in their room. The answer was negative and this was pointed out to be a possible reason behind the error in scaling in freshmen.

When asked if they felt that they had improved their presentation skills in this period, they all agreed to this thought and explained in detail how they had to put more effort into preparing proper presentations in order to be able to describe and present their projects and work-in-progress in the digital environment. They have explained that it was also expected of them to prepare these presentations for their work-in-progress by their tutors in order to efficiently communicate on their projects, which wasn't expected of them in a conventional studio environment. However, even though they were aware that this period had improved their presentation skills, they felt that the increased amount of effort became unbearable at times.

Regarding the problems that the lack of individual perspective during presentations brought in distance education, it was revealed that tutors did indeed use groupboard software such as Miro for critiques. However, the tutors demanded full-screen slide presentations, controlled by the students for midterm and final juries. Students stated that they preferred to use Miro, where everyone had their own views on the presentation material and that the communication in such an environment was much more efficient, rather than tutors demanding they go back and forth between slides and request zooming on certain points of the material from the presenting student.

On the issues about layouting and the quality of the presentation material, students stated that having been preparing such material for the whole semester made them want to spend less and less time on this task, thus a decrease in the quality of the presentation material over time.

It is worth noting that when the scope and the intent of this research was finally explained, the students immediately suggested that it should include VR capabilities. However, in a later, informal discussion, when further asked if they ever had the chance to try such VR software or owned a VR headset, it was revealed that none of the students had neither used any BIM or AEC software in VR nor owned a headset, and that their interest in VR was mostly based on the current “Metaverse Hype”.

When asked if they preferred to have separate software for certain tasks or if possible an all-in-one solution, they stated the same concerns as the tutors, that too many windows could become unmanageable and impaired their communication and concentration. Adding that they would prefer an all-in-one solution if it properly answered all of their needs during the classes. Discussing the possibility of such software, they stated that if the pipeline was simple and fast enough, they would prefer to have a fast system rather than a half-baked all-in-one approach, giving the example of the Adobe ecosystem where every software has their own use but are easily imported/exported between each other.

3.4. Findings

Upon reviewing the survey and the interviews, various topics in three categories were extracted from these preliminary studies to group and better define the findings. Defining the categories of these findings is important in outlining the gathered information. The information gathered can be simply categorized into these three categories: Baseline, Advantageous, Disadvantageous.

The “Baseline” category contains information that is neutral and foundational in nature, and may not be relevant to the distance education period. Topics such as students’ design and presentation habits, their certain preferences and concerns, department-associated similarities and differences, and basic necessities fall under this category. Some of the information here will only be pointed out rather than discussed, acknowledging that such topics are not within the scope of this research and may be topics of research of their own. The information in this category will be used to define the most basic necessities of an architectural digital design studio.

The information under the "Advantageous" and "Disadvantageous" categories are purely based on the experiences from the distance education period. Information under "Advantageous" will be inspected to reveal the underlying factors to be able to reproduce and perhaps strengthen such possibilities. On the other hand, the topics under "Disadvantageous" will also be inspected in the same way to be able to suggest solutions to prevent such issues from happening.

These findings, especially the ones that fall under the latter categories will also respond to the first research question (Q1), which is "*What are the challenges faced by traditional architectural design studios in the distance education model and what are the opportunities this model offers?*".

3.4.1. Baseline

- **Basic Design Studios**

First of all, it is important to remind that all students of the Istanbul Bilgi University Faculty of Architecture must attend "Basic Design I" and "Basic Design II" studios regardless of their department. In these first-year design studios, students start to work with smaller physical models, which get bigger in scale over time, and students eventually proceed to fabricate 1:1 scale models, often in the form of pavilions. For the duration of these first-year studios and with other supporting classes, they also learn how to use Rhinoceros as a design tool, especially for 3D modeling and fabrication.

From both the students and the tutors' statements, it was revealed that these first-year studios contribute to building their scale perception through these physical models.

Working with physical materials is also important in these studios, as students also get the opportunity to learn about material qualities and have the chance to use and test them in various scenarios.

During these studios, through the lectures given on the usage of tools and machinery provided in the fabrication laboratories of the university, students get the chance to meet workshop machinery and digital fabrication techniques such as CNC routing, laser cutting and 3D printing, which they often use throughout their education for fabricating their physical models.

- **Specialized Design Studios**

After the first year, students attend separate design studios associated with their departments. In these department-oriented design studios, students start to differ from each other as they get more specialized in their fields. The studios themselves are also considerably different from each other. As an example, architecture students tend to design larger projects as they advance, which isn't the case with the industrial design students, who tend to design much smaller objects compared to architecture students, regardless of their year. Approaches and methods also remarkably differ between these studios, where architecture students tend to go more and more digital over time, doing smaller scale and fewer physical models and more detailed 3D models and renderings, industrial design students tend to spend less effort on digital drawings and more on fabrication methods and physical models, which are usually in the form of 1:1 scale prototypes.

It is worth noting that these are not the only differences between the design studios in separate departments. Even though they may have similarities in their environment and some aspects, they are extremely different in syllabus and requirements, and should not be generalized.

- **Software Usage and Digital Media**

Due to the difference between the scale and the scope of the projects in the design studios of separate departments, students show a tendency to use specialized CAD software for their needs. Even though all students learn Rhinoceros in Basic Design Studios and the tutors expect them to continue to use it, students do not stick with Rhinoceros. This implies that if a common data type was to be selected for a reason, it should be a universal type of data where most of these software can export.

While the most commonly used modeling software among students, Rhinoceros, works with NURBS geometry, the second most common modeling software Sketchup uses mesh geometry. Keeping in mind that game engines only work with mesh geometry, this information also implies that a geometry pipeline has to be thought of regarding Rhinoceros users when a game engine is being utilized.

Static images, animated GIFs, videos and PDFs should also be considered regarding a digital design studio, as these types of data are also found to be commonly used by the students.

On a common ground in software usage, all students are acquainted with pan and orbit controls, which is prevalent in CAD software. Despite not all students playing PC video games, first-person controls were also not foreign to them and scored almost as well as (3.48) pan and orbit controls (3.79) in how comfortable they felt using, in the survey. Taking that they all know how to use Rhinoceros into consideration, first-person controls scored higher than expected in comparison to pan and orbit. This was tied to the fact that most students utilize rendering engines, such as Lumion, and that these engines usually have first-person control schemes for navigating through a model.

- **Physical Media**

It is worth noting that students also tend to produce physical sketches and models regardless of year or department. This has been the case even in the distance education period. This suggests that a digital design studio should always seek to provide support for such material. It is and should always be in the students' preference to choose their media for both designing and presenting, even when the environment is completely digitized, the environment should be responsible for supporting all kinds of media.

- **Design Development**

Generally speaking, regarding the methods used in various stages of the projects in terms of design, it was confirmed that students tend to move from sketches to 3D models as their project progresses. In correlation, utilized representation material and methods evolved from sketches and diagrams to 3D views. However this statement should not be used to generalize all of the students' behavior as there is a fair amount of students that do not follow this path. It is also worth noting that some studio tutors have their own requirements for the methods and media to be used in their studios.

- **Synchronous Interaction**

The conventional studio space is a place for synchronous collaborative working and interaction. During the distance education period, after using environments that have such capabilities, both the students and the tutors preferred and have seen advantages in them. Students prefer to work together for group projects in such environments when possible, where they can more efficiently collaborate.

- **Work Privacy**

Although it was not discussed thoroughly in the preliminary studies, students can tend to be secretive of their work. The reasons behind this secrecy can vary from insecurities about the quality of their work, to concerns about plagiarism and competitiveness. In a conventional design studio, students can often discuss their work with their tutors privately and not have to present and have it seen by others besides juries. In the distance education period, and through some practices held by the tutors in both situations, students have lost their ability to keep their work private. This issue can severely impact students' participation and attendance to classes.

3.4.2. Advantageous

- **Efficient, Smaller Groups**

Resulting from the inability to work in larger groups, students had to form smaller groups than usual. Compared to the previous, face-to-face terms, tutors found that these smaller groups were much more efficient and produced better results.

- **Virtual Site Exploration**

The disadvantage of inability to physically visit sites, resulted in the increased interest in virtual exploration. Through using GIS and similar software, students showed a tendency to better understand their project sites through other means.

- **Online Presence**

Online presence has brought many advantages as well as disadvantages, of which the latter are explained under "Absence". The first of these advantages, on a smaller scale, is the elimination of distance in a physical space, as in being able to see everybody in a session at the same time. Taking this topic to a larger scale, online presence has also eliminated the larger distances, as in participants joining in from different cities or even countries, without the need for travel. This could also be interpreted as an advantage in travel costs and saving time.

- **Better Presentations**

From the lack of a physical studio space, students have felt the need to properly prepare presentations for their material each time they were to discuss them. Students would show their work-in-progress material to the tutors in the conventional studio space, yet they did not prefer to do so in the online sessions. This situation caused students to improve their presentation skills.

- **Recorded Sessions**

Students found the recorded sessions to be very helpful in this period, which was uncommon in the conventional design studios. Even in face-to-face conversations, students could miss or forget about some points discussed with their tutors and having recording of these discussions helped them through being able to revisit them. On a related note, students also found the permanent methods of discussions and annotations to be more effective as they can be stored and viewed later. Verbal communication and temporary drawings tend to be forgotten if the sessions aren't recorded.

3.4.3. Disadvantageous

- **Unsatisfactory Basic Design Studios**

Due to the fact that Basic Design Studios are conventionally based on physical model-making approaches, they were impacted severely due to the inability to reproduce this environment in the distance education period. What could be reproduced wasn't found to be satisfactory both by the students and the tutors. Trying to follow a syllabus that was created with a physical studio space and the university's facilities in mind was found not to be ideal.

- **Lack of Fabrication Lab Access**

Adding to the previous topic, inability to access the fabrication laboratory impacted freshmen the most. Being unable to use the lab, freshmen students couldn't produce bigger scale models as they would, in addition to not learning about how to operate the tools and the machinery.

- **Impaired Scale Perception**

In the case of freshmen students, directly related to not being able to produce bigger scale models, Basic Design Studios' students missed a part of their education where they would gain proper scale perception. It was also found difficult to communicate on scale on a screen even in face-to-face education, and even more so on a digital medium such as a video conferencing software.

However, this issue did not only affect the freshmen. Students from all years were found to have impairments on their scale perception, more than usual in this period. There appears to be several underlying factors that caused problems in scale perception, which are explained in the other topics.

- **Lack of a Studio Space**

Affecting all studios from all departments and years, the lack of a studio space impacted both the students and the tutors. The studio space is found to be very important in the Faculty of Architecture. It could be simply described as the space where the studio sessions are held and a common working place for the students. However, this would be a shallow definition. Apart from the studio sessions, the studio space provides an environment for the students to learn from each other and socialize, regardless of their studio groups and years. Not having a physical common space in the distance education period has impacted the students' opportunity to interact with each other. Even though the students who knew each other from before continued to communicate online, the lack of a common space impaired the students' opportunity to form new relationships. This impact has affected not only their work but also them, socially and psychologically, as it was stated both by the students and the tutors. The lack of a studio space was stated to be the reason behind many other issues such as communication efficiency, concentration problems and scale perception.

On the issue of concentration, the studio is thought to provide a working environment in an isolated manner, where the students can focus on their work without distractions that they may have in their homes.

Freshmen were particularly affected by these issues. Working in a studio space persists throughout their education, and it is considered to be a habit to be gained in the introductory Basic Design Studios.

- **Lack of Field Trips**

Distance education and pandemic conditions also brought the lack of field trips. These visits to project sites are found to be important both by the tutors and students in understanding the qualities of the sites. This is also thought to be one of the reasons behind the problems of scale perception as the students couldn't experience the site to perceive the actual dimensions of it.

- **Physical Absence**

On the contrary to the advantages that online presence has brought, physical absence has brought many disadvantages. First of all, combined with the lack of a physical studio space, students have nearly lost all interaction due to not being physically together in any space. This was thought to have influenced the students to work individually, more than usual.

Participation in class was also affected due to online presence. When not required, students tend to turn their cameras off and refrain from participating in class. Keeping their cameras on being regarded as the most common way of showing presence, absence could only be prevented by doing so in the current distance education environments. The feeling of absence both impacted the students and the tutors negatively. As students' participation fell, tutors' motivation also seems to have suffered.

- **Concentration Issues**

Without feeling presence, students had concentration issues. Presence through keeping cameras on was the only way for some students to stay concentrated. Without small breaks and informal conversations during classes, students lost concentration when they felt overwhelmed. Being in some sort of a studio space would make it possible for them to take small breaks and have more freedom, instead of the constant video conference stream of a studio session with only a single conversation being held at a time.

- **Communication Inefficiency**

An inefficiency in communication was felt by both the students and the tutors. Annotation and drawing tools provided by the used software and environments were utilized. However, the lack of other usual methods of communication such as gestures, pointing or using the space and the physical objects around was felt.

Being unable to communicate on physical models through physically reaching them was also found to be a significant problem in the studios where physical model-making was important.

- **Technical Difficulties**

From inability to efficiently use distance education software, to having too many windows open at once, technical difficulties and distractions were felt and impaired the whole process.

- **Lack of Individual Perspective**

Could be perceived both as a technical difficulty and a reason for communication inefficiency, not having any control over the presented material was found to be a major problem. Groupboard software such as Miro were used where participants can have their own individual views on the shared material, but the problem persisted through not utilizing such environments at all times. An environment where users could simultaneously view a 3D model was not sought, thus not experimented with.

Whether sought or not, a digital design studio should provide the opportunity for the participants to control their own view on the shared material.

- **Loss of Senses**

Mostly concerning Industrial Design Studios, material qualities were limited to visual and auditory in this period. Qualities concerning other senses couldn't get across to the other side as there is no substitute or a way to communicate them in today's technology.

- **Increased Influencing**

Concerning recorded sessions and shared submission folders, students felt that having the opportunity to view other students' work so easily resulted in increased influence between students' works. Tutors have also shared this concern by stating that the "risk takers" have decreased and the tendency to stay inside comfort zones grew among students. This could result in less diversity in students' work and a decrease in effort and quality. Without eliminating the advantages of recording sessions, some technical steps could be taken to prevent this issue from happening.

3.5. State-of-the-Art

Keeping the previous research and the initial findings in mind, a review on the state-of-the-art, as in existing software including video games, was conducted. The reviewed software are not necessarily related to architecture, education, remote working or multi-user virtual environments. The review was done in a way that was focused on the opportunities that these software may bring to the field of architecture, distance education or simply the features that the design studios may benefit from.

This review was done with aid from the author's previous experience on such software and a library of approximately 1000 video games. As most features and methods discussed are present in many of these software, only a selected few with prioritization for more recent ones, were revisited to prepare this report.

The features, or opportunities, within these software were defined under topics, and will be explained regardless of the software that employ them. The purpose of this review is to examine the features themselves, not the software that employ them.

- **Avatars**

In most video games and multi-user virtual environments, regardless of the genre or purpose, users are represented by an entity or a group of entities. In groupboard software or collaborative online documents, these may come in the form of pointers or carets, while in video games, avatars and controllable objects are commonly utilized.

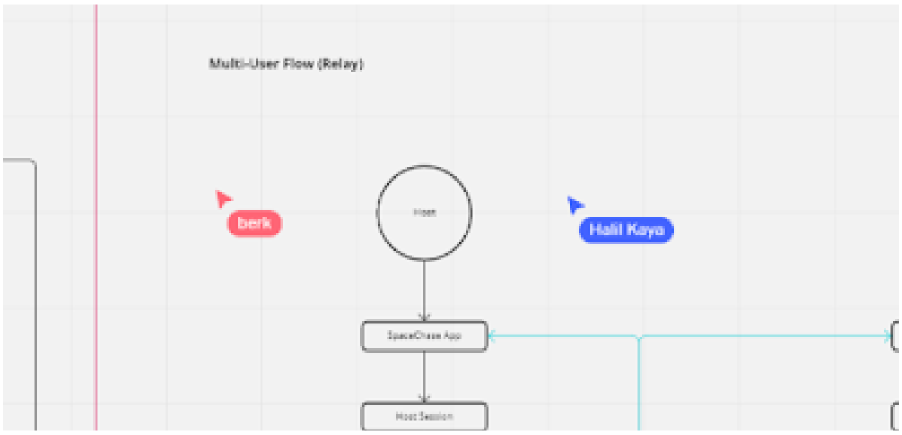

Figure 3.27. Pointers of Other Users in Miro

These avatars usually contain an indicator of the user's identity in multi-user environments, usually in the form of a name tag which can either include their real name or a chosen nickname. On the software where a single entity is controlled, such as a humanoid avatar, customization options are usually utilized to let users distinguish themselves.

Figure 3.28. Wardrobe in Cyberpunk 2077 (URL-10)

In software where avatars are used, the user control schemes, regardless of the scheme itself, almost always control the avatar. In situations where users control various objects, such as RTS games, the users are not represented by avatars but are rather controlling their units separately. In such cases, awareness between users is decreased and only the consequences of decisions can be observed. This is also the case in top-down, turn-based RPGs, where users control characters but their cameras are detached from their avatars. This situation indicates that in order to increase awareness between users, cameras attached to avatars can be utilized as their actions regarding their view of the environment and camera controls can also reflect on the avatars through animations.

- **Animations**

Regardless of whether users control avatars or not, animations are largely utilized in video games. These animations can be related to animated objects, mechanisms, visual effects and avatars. In the case of avatars, animations can reflect their movement, actions or gestures such as social ones.

Figure 3.29. Players Dancing in a Deep Rock Galactic (URL-11) Lobby

Procedural animations in combination with inverse kinematics can be used to realistically reflect certain scenarios, such as walking on rough terrain or the avatars heads rotating as the users’ cameras rotate.

Figure 3.30. Foot IK in Grand Theft Auto Online (URL-12)

The same approach can also be used to develop further procedural animations, such as pointing and seating.

Figure 3.31. Pointing in Grand Theft Auto Online

- **Camera**

In a virtual environment, the virtual camera is the most important object in rendering the environment for the users. As it is virtual and programmable, various perspective modes and properties can be programmed to achieve certain views and visuals. Perspective and orthographic cameras are commonly used in such environments. As the cameras are programmable, these camera modes can be switched instantly if the environment has implemented such options. Camera properties such as position, rotation, focal length and field-of-view are also easily programmable. VR modes are also directly related to camera programming as they require the same scene to be rendered for both eyes, from different positions.

Figure 3.32. TPP-FPP switch in Grand Theft Auto V (URL-13)

Regarding avatar usage, and the previous remarks about cameras being attached to avatars, third-person and first-person perspective modes are most commonly used in combination with avatars, often with the ability to change between these modes. Commonly used in round-based video game modes, users can also spectate other users' cameras when they are eliminated and waiting for the next round.

- **Cross-Platform**

In video games, the term cross-platform is usually related to users being able to join in on a game session from either PC or gaming consoles. On multi-user software focused on collaboration however, this can indicate support for various hardware other than consoles.

Having users connect through their PCs, tablets or phones is regarded as a form of cross-platform capabilities. Further adding to these capabilities, the possibility of having VR and AR users in addition to the ones using screens connecting with each other also became a form of cross-platform approach. Today, many video games exist where VR and PC users can cooperate, with some games taking the interesting approach of differentiating the role of separate platforms, such as Davigo, where PC users work together to try and defeat a giant VR user.

Figure 3.33. PC vs VR in Davigo (URL-14), Image (URL-15)

Some software and video games also feature various types of clients that run on different platforms. Spatial for example, connects VR users from an installed client and PC users from their internet browsers. Unity is known to provide such opportunities as building applications for a large variety of platforms including PC, Mac, Linux, VR headsets and WebGL.

Figure 3.34. Spatial Running on an Internet Browser

- **Communication Methods**

For communication inside multi-user virtual environments, methods involving text, voice and video communications are commonly utilized. For text communications, chat boxes and leaving comments can be given as examples. For asynchronous communication, comments and suggestions can be left, which are frequently used in collaborative working software. For synchronous communications, real-time chat boxes, private messages, real-time voice and video communications are regularly used. The communications can also be filtered through channels and groups. By forming parties and groups or connecting to separate channels, users can hold simultaneous conversations separately from each other. Proximity chat is also widely used in cases where the environment holds many users, through this method, only the users that are within a certain distance of each other can hear others. These methods can be used together as well, such as selectable channels in chat boxes or binding certain keys to separate channels, such as global and proximity voice channels.

Figure 3.35. Voice Chat Bindings in Phasmophobia (URL-16)

- **Real-Time Interaction**

Unlike consecutively worked documents and asynchronous collaboration, real-time multi-user environments can also provide real-time interaction, naturally. Users can see changes as they happen in these environments, which is necessary for multi-user approaches aiming for workspace awareness.

One of the most important aspects in real-time interaction though, is the consequential awareness that it can provide. For example in first-person shooters, users can see the trajectory of their projectiles, or when their teammates or opponents draw and aim their weapons before they shoot, and in crafting/survival games, users can often see what themselves or the other users are about to construct as they are deciding.

Features regarding consequential awareness can help make decisions in combination to the workspace awareness that a collaborative environment can provide. In real-time environments, consequential awareness can reinforce collaboration and decision-making.

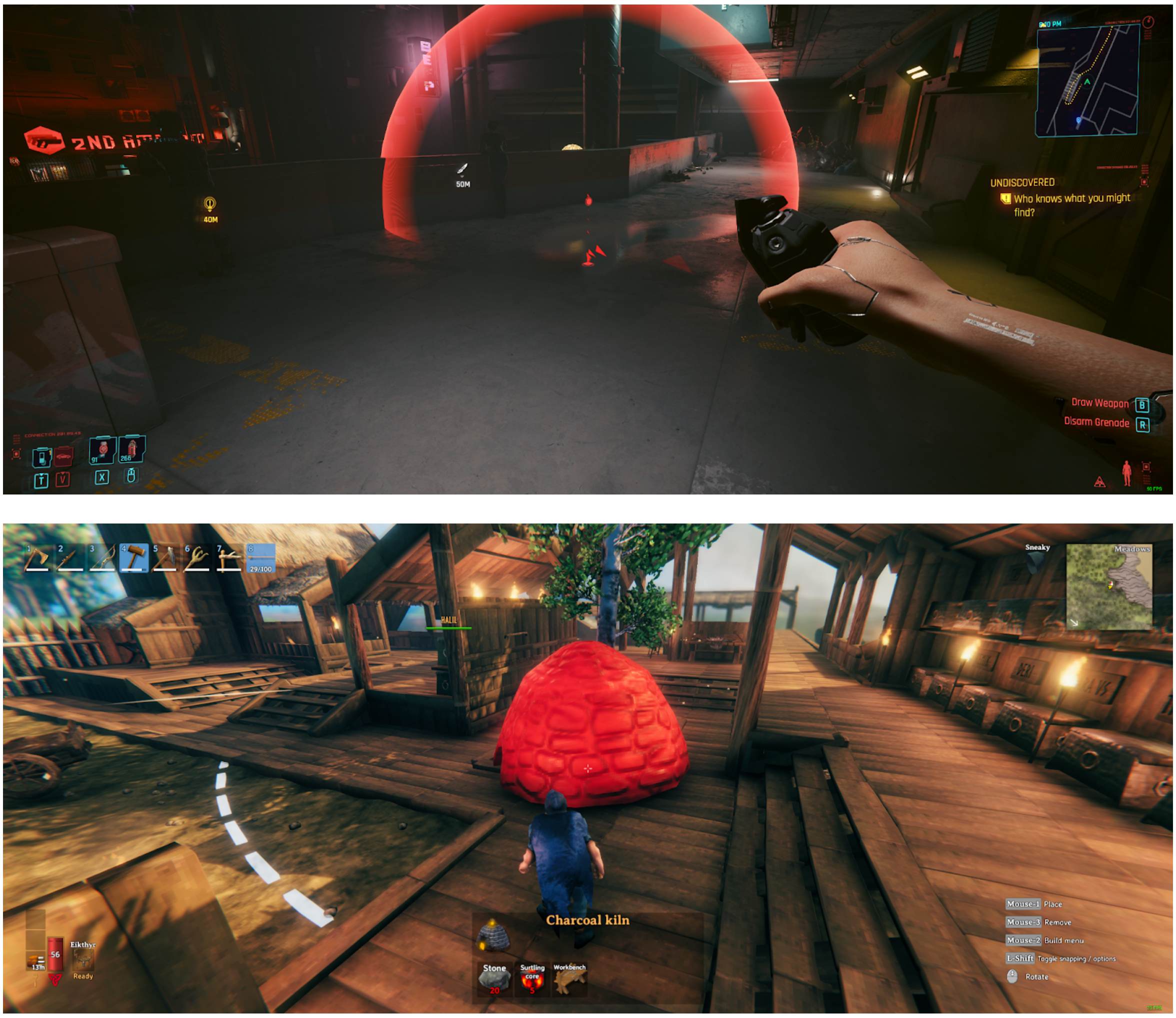

Figure 3.36. (a) Grenade Trajectory in Cyberpunk 2077

(b) Invalid Placement of a Charcoal Kiln in Valheim (URL-17)

- **Recording**

Mostly in racing games, or competitive games featuring replays, sessions can be recorded. The discussed recordings are not done through video records, but through action recording with vectoral logs. These types of recordings can let users view the entire session through views that weren't rendered during the real-time session, thus providing an opportunity to analyze and see things from different perspectives.

- **Networking**

In multi-user environments, several methods are commonly utilized. These can be grouped in two, centralized and decentralized. Centralized approaches depend on a host or a server assuming the duties of synchronization between users. Dedicated servers, online services and host-client approaches fall under this category. On the other hand, decentralized approaches rely on the users to handle the connections and synchronization between themselves. As an example, P2P schemes have all users being connected to each other rather than one entity controlling all the data, thus the session persists as long as there's a user still connected to it.

- **Sessions and Lobbies**

Directly related to networking, is the topic of sessions and lobbies. Sessions are instances of an environment, a software or a multiplayer video game can have many sessions that are not related to each other in any way. Through utilizing sessions, users can group up and cooperate in these environments, or meet with other users in public sessions. Lobbies however, which are often synonymous with sessions, are the preliminary environments to sessions. In most video games with linear progression, users meet in a lobby before starting their session. In video game lobbies, users can set up the session, adjust their equipment and characters, communicate and discuss tactics.

On the other hand, in MMOs, crowded sessions with up to thousands of players are maintained. In these kinds of games and virtual environments, multiple servers support a single session and the session is kept live unless for maintenance and updates. However, in more private game modes or certain instances in these MMOs, where it isn't appropriate to have an open session, private instances are created in parallel that hold a certain number of users, such as dungeon instances or PVP arenas. Hundreds of groups can be in the same dungeon without ever seeing each other through utilizing these parallel instances. Although this doesn't necessarily mean that separate servers are hosting such instances, a single server can filter the synchronized data by user IDs and keep groups from seeing each other.

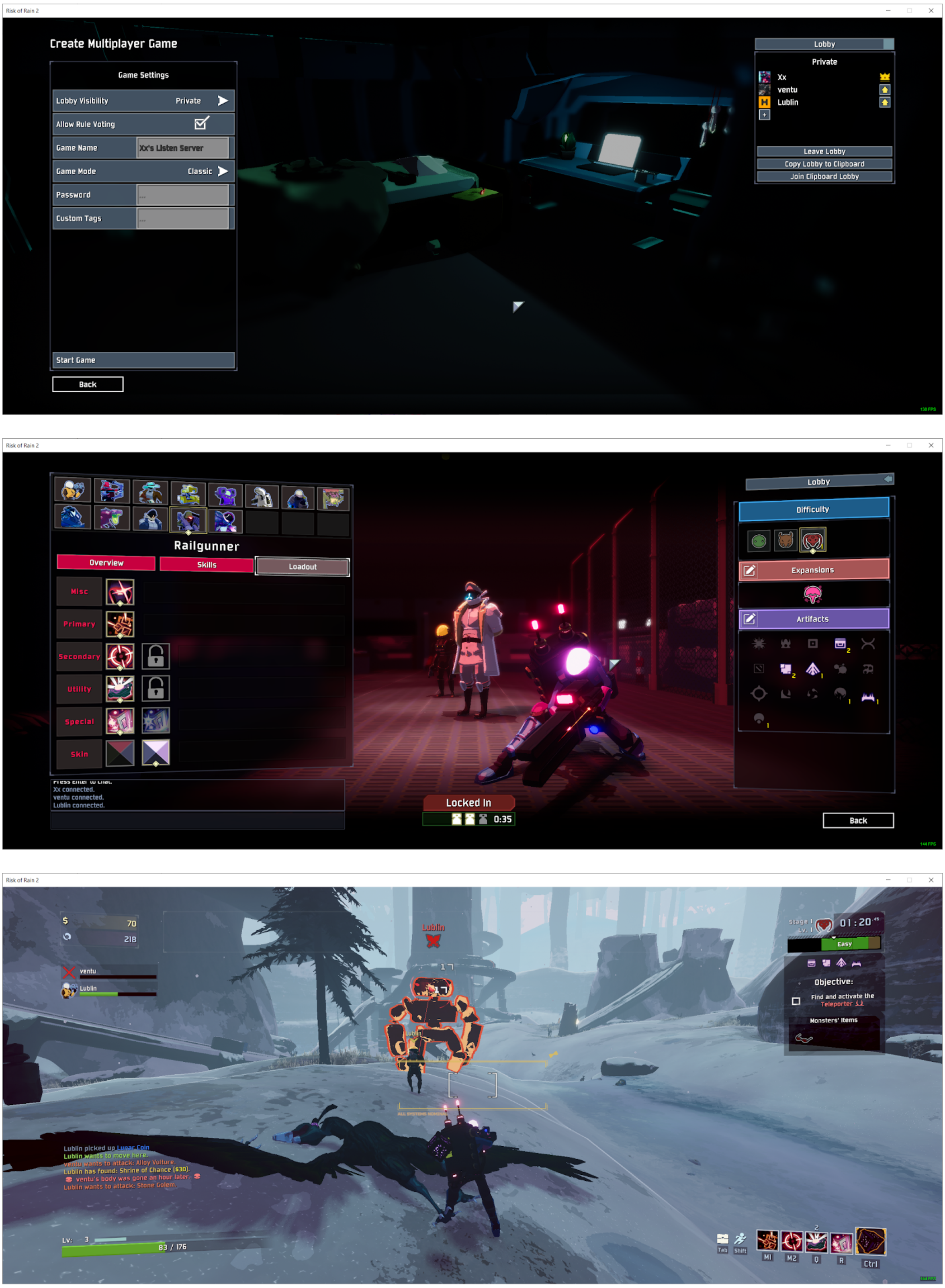

Figure 3.37. Various Stages of a Multiplayer Game in Risk of Rain 2 (URL-18)

- **Importing Data**

In video games, data importing and exporting is not very common. Custom maps and player models can be added to some games through either mods or official support but that's usually as far as importing goes. Runtime importing is rare and usually in the form of smaller data types such as images and textures for custom player models or spray tags.

Multi-user collaborative working or social environments however, often include support for runtime importing on various data types. Images, flippable PDF documents, static and animated models are supported import types in environments such as Spatial. Bidirectional pipelines, where the changes on the imported material can be synchronized between the environment and a CAD software are more common on collaborative working platforms aimed for the AEC industry, such as Arkio (URL-19).

One important factor in importing models is the support for colliders. As most of the collaborative environments aimed for the AEC industry are also focused more on VR hardware, colliders are of less importance and thus, not supported. The necessity of colliders, and what they are, will be explained later in a more technical perspective.

Figure 3.38. Importing 2D and 3D Objects in Spatial

- **Prefabs**

Prefabs, short for “prefabricated objects”, are common in all kinds of environments whether a video game, social environment or an online collaborative working space. These objects often come in a catalog and are ready to be placed inside the environment by the user. Rendering engines utilize prefabs for objects such as trees, NPCs, vehicles and clutter. In video games however, prefabs contain further capabilities rather than being static visual objects. For example, in survival and crafting games, constructing buildings require the use of prefabs which may have joinery conditions, snapping points and even structural integrity. In the Viking-themed survival sandbox Valheim, structural objects can break if they aren’t supported in a semi-realistic way, the further an object is from the ground in terms of order, it becomes weaker and eventually rejects being placed.

Figure 3.39. Building Prefabs in Valheim

Parametric geometrical objects are also utilized in environments such as Second Life (URL-20) and Arkio. These objects often come in the form of basic geometrical shapes. Such objects can also be manipulated through their control points, edges or corners, and even be subtracted from or merged with each other, through performing boolean operations.

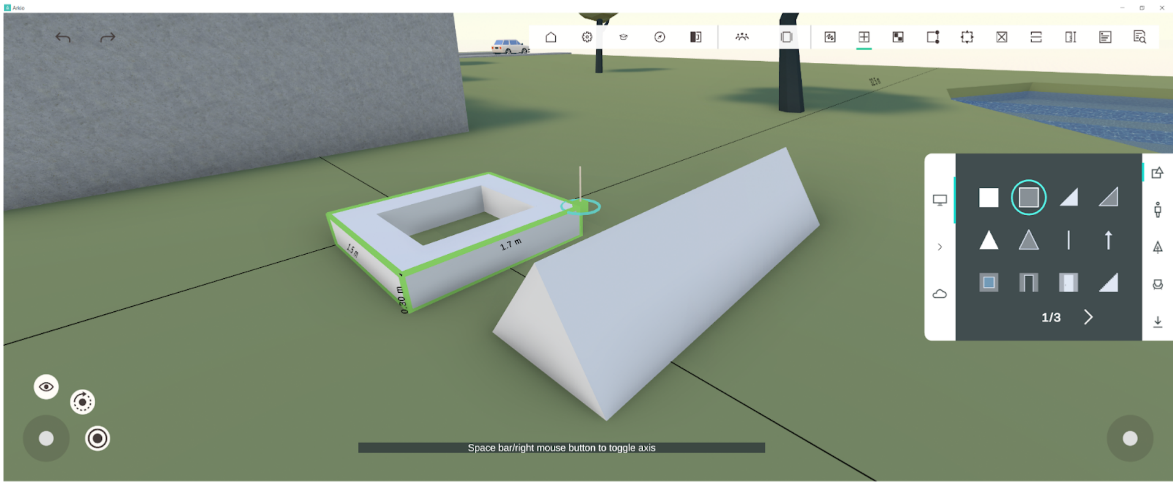

Figure 3.40. Parametric Objects in Arkio

- **Annotations**

In most of the online collaboration platforms, some form of annotation objects exist. These annotation objects are often in the form of sketching on a plane, or placeable text boxes. A screen space overlay can be a plane for such drawings, as it is on the video conferencing software Zoom. However, in document-oriented collaboration platforms such as Google Drive and Google Docs, these annotations are in the form of comments that are hidden from plain view.

Figure 3.41. Participants Annotating in Zoom

On the other hand, in video games such as Deep Rock Galactic, where spatial communication is important as the players are navigating through a procedurally-generated cave, 3D oriented annotation tools are being utilized. The laser pointer is one them, where users can mark a point for the other users to see, which also contains information about the marked point and can have programmed reactions in the form of phrases voiced by the characters to some objects, such as shouting the type of the marked enemies or hilarious reactions to large nuggets of gold.

Figure 3.42. Laser Pointers in Deep Rock Galactic

- **Maps and Minimaps**

Whether in 2D or 3D, in most of the software containing large environments, maps, minimaps and compasses are utilized. In these environments, minimaps are often placed in the corner of the screen, showing the immediate vicinity of the player, accompanying a compass indicator. In most of the video games with maps, a wayfinding feature is also implemented to guide the player to their destination.

Maps and minimaps are crucial in orienting users in large environments, several other methods such as large landmarks, an east-to-west rotating sun, and compasses are often utilized when maps are absent.

Figure 3.43. (a) 3D Map and Routes in Cyberpunk 2077

(b) Minimap in Cyberpunk 2077

- **GIS and Satellite Data**

Software such as Arkio and Microsoft Flight Simulator X, utilize features regarding importing GIS and satellite data. Through the use of such methods, real world locations and geometries, such as buildings, can be imported into the environment. The OpenStreetMap (URL-21) project supports such programming through its API.

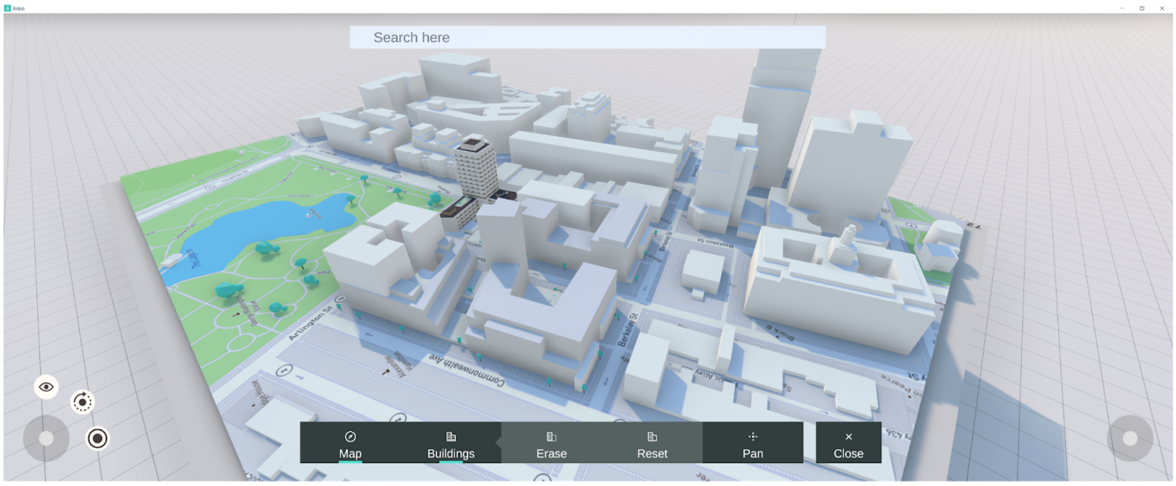

Figure 3.44. OpenStreetMap Import With Buildings in Arkio

- **Analysis and Simulation**

In all virtual environments, as the environment itself is a product of programming, endless types of analysis and simulation methods can be implemented. These methods are often seen in the "Simulation" genre of video games. The city building video game Cities: Skylines (URL-22), for example, features 36 analysis views for the cities built inside the video game, displaying various layers of information. The "info-views" in Cities: Skylines include but are not limited to electricity infrastructure, water and sewer infrastructure, citizen happiness, traffic, noise pollution and land value. Most of these analyses result from the simulations within the video games.

In analysis software or CAD software equipped with analysis capabilities however, these analyses are done on a professional and accurate level. Although still being a product of programming, these methods can actually be implemented to other virtual environments.

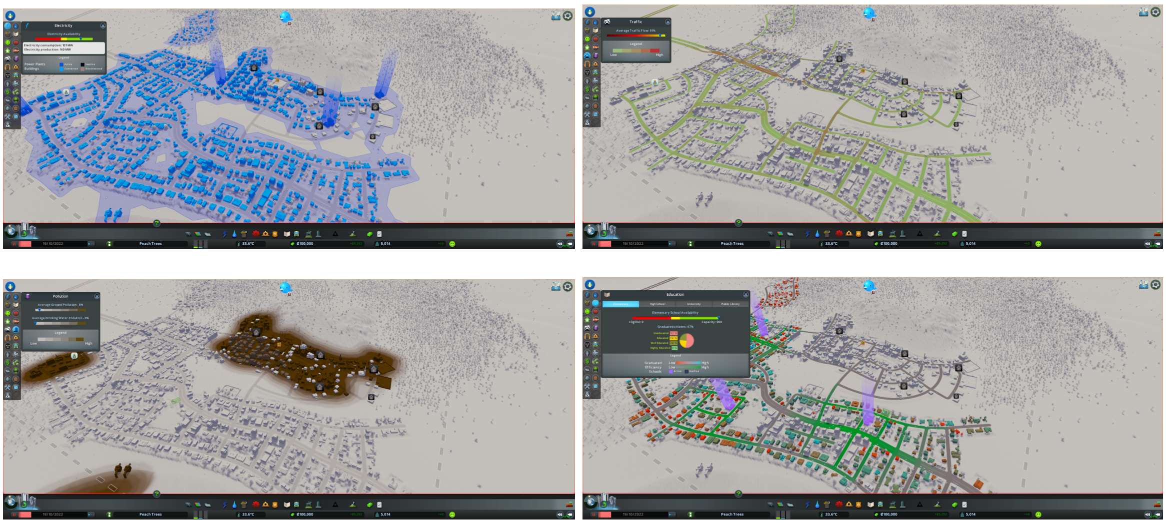

Figure 3.45. Various Info-Views in Cities: Skylines

- **AI and NPCs**

Utilized in different scales for rendering engines and video games, NPCs can populate the virtual environments. Video games often utilize NPCs as a part of their narratives in addition to being environmental objects. Such NPCs can have programmed day and night cycles, behavior and reactions. AI plays an important role in NPC programming inside video games, though not as much in rendering engines.

Figure 3.46. A Street Crowded with NPCs in Cyberpunk 2077

Pedestrian and vehicle traffic is often simulated with these NPCs, which helps create lively cities and environments inside video games.

- **Lighting**

Various lighting methods associated with the environment are present in video games. Realistic day and night cycles can be achieved through such methods. Ray tracing is also utilized in more recent games, which simulates the physical behavior of light.

In more professional software, sun path algorithms are utilized to analyze lighting and shadows in a real-world situation. In addition to the sun, artificial interior lighting and daylight factor can also be calculated through such software and programming.

Figure 3.47. Day and Night in Night City, in Cyberpunk 2077

- **Shaders**

In addition to lighting, certain programming related to rendering called shader programming, or simply *shaders*, are utilized to produce visual effects. Shaders can overlay the actual geometrical information through manipulating the visualization of the geometries or the rendered pixels. X-Ray vision, dithering, edge detection, cross sections, and weather effects such as vertex displacement snow shaders can be given as examples to shader programming.

Figure 3.48. Cross Section Shaders by Abdullah Aldandarawy (URL-23)

Figure 3.49. Snow in Monster Hunter: World (URL-24)

4. FEATURE LIST

Following the preliminary studies and the state-of-the-art review, a list of expected features of a digital design studio began to form. Starting with an unfiltered list of the requirements, deriving from the findings of the preliminary studies, these requirements will later be filtered in terms of technical surmountability. Through discussing the technical aspects of such requirements, some compromises will have to be made, which will be explained further. After making the decisions regarding technical constraints and surmountability, additional features, which do not derive from the requirements but rather emerge from the possibilities, will be discussed. Finally, through outlining this expanded feature list, the technical design of the features will be made and translated into a table that will be referred to as "The Feature List", which is a common document in developing software. This list will help with the prioritization and grouping of the features in terms of development, and will be updated regularly throughout the development process.

Starting with the first step, the following requirements were extracted from the preliminary studies.

- **Multi-User Capabilities**

While it may seem like an obvious requirement, the extent of these capabilities are more critical than it may be perceived. It is within the limit of such capabilities to let certain forms of synchronization occur. In a digital design studio, it is suggested that these capabilities should be in real-time and offer the opportunities for collaborative working.

However, this shouldn't mean that the only form of communication must be in real-time, there were certain implications that brought the topic of asynchronous communication and collaboration, which is determined to be an important requirement in certain scenarios.

- **Presence**

Directly related to the multi-user capabilities, the sense of presence and workspace awareness was found to be equally as important. Physical presence being commonly substituted by cameras in the current distance education methods, the state-of-the-art research proved that there may be other methods of showing presence, such as avatar usage. Whether in the form of mouse pointers in a 2D groupboard software or humanoid avatars in a 3D virtual environment, another method of participant presence could and should be achieved rather than being constrained to camera usage. This should also be done in a way that provides workspace awareness, in real-time.

- **Digital Studio Space**

A common space that could simulate a physical design studio was found to be an important requirement for a digital one. This environment should provide the opportunity for the students to meet and communicate in separate groups during and outside of studio hours. This implies that such an environment should always be accessible and be able to be repurposed when necessary. Due to the nature of all the design studios in the Faculty of Architecture, this environment should have volumetric qualities to provide support for 3D models. However, this doesn't mean that only 3D models should be supported in such an environment, 2D media are as important as 3D ones and should have equal amount of support if not greater.

The requirement of a 3D space should not only be perceived as an obligation for having support for 3D media. Combined with a proper method developed for presence, this could bring the advantage of providing the means for better scale perception among students. As an example, a 3D environment where the students can upload their models and have walkthroughs with a reference to an actual human size projected in the correct scale, students could have the chance to properly perceive the spatial qualities and the scale of their work or certain spaces.

- **Data Support**

Through conducting the preliminary studies, it was understood that a digital design studio shouldn't mean that the working material will also always be digital. This brings us to the conclusion that such an environment should support the preferences of the students and the tutors, as in having support for a wide range of data formats to accommodate the presentation of all kinds of material, such as 3D models, vectoral drawings, digitally produced images, high resolution photographs, videos and multi-page presentations. It should also be considered that in the possible event of a repurposed digital studio space, perhaps for a digital field trip or a jury setup, large amounts of data should be able to be seamlessly transferred and used.

On the topic of data privacy, students should be able to have control over the visibility of their work, even in an always-online digital space.

- **Communication Tools**

It was determined that a digital design studio should have spatial communication tools as well as written and verbal. Where hand gestures and pointing is utilized in a physical environment, it was understood that such gestures didn't efficiently get across in a video conference session and some were lost completely. Having 3D communication tools could substitute and even improve such communication beyond conventional methods.

In the event of having a 3D space and 3D models, 3D annotation and drawing tools should be also developed. As digital 2D annotation tools were found to be efficient during this period, an improvement and further development on such tools should be sought, especially reminding the support on 2D material.

The permanence of communications held in a digital design studio was found to be advantageous and should be an important requirement in designing one. Improving on the temporary annotation tools, such as drawing on the projected screen, ways of storing such interaction should be developed. Means of storing verbal comments could also be developed for efficiency, such as speech-to-text or audio clips, rather than relying on the complete recording of sessions.

- **Tangible Model-Making**

Concerning mostly the syllabus of Basic Design Studios and the specialized studios where model-making is preferred or found crucial, methods of simulating the interactions of physical model making should be developed. The interactions sought were defined as: being able to hold and rotate a model freely, manipulating the said model through subtracting, adding or rearranging pieces, and understanding the structural integrity and the material qualities. Additionally, more experiential aspects tied to auditory and visual qualities should also be considered such as lighting and shadows, light refraction and transparency. However, regarding other senses, qualities such as roughness, elasticity, scent and thermal properties should be considered but they were acknowledged not to be probable regarding the current technological possibilities. It is worth noting that some of the behavior regarding these qualities can be visualized, but the qualities themselves cannot be simulated yet.

- **Virtual Fabrication**

Aimed for learning about the fabrication technologies and how to safely operate these tools, fabrication methods and operations should also be simulated for the students. This suggestion however, depends on the assumption that the digital design studio conditions are either temporary or used in combination with a physical setting, as it would be unnecessary for the students to learn how to operate such tools if they were never going to have access to them. This could also bring the advantage of having safety training in a completely safe environment and then actually operating in the real environment and tools. Although desirable for safety training and learning about fabrication principles on a basic level, it should be understood that this couldn’t be a substitute for situations such as craft workshop internships. However, simulation of fabrication could be taken a step forward on a larger scale, where it would be possible to simulate the construction of an entire building.

4.1. Filtering and Constraints

In the scope of this research, the approach for a digital design studio was chosen to be more focused on design communication, rather than the act of designing or its fabrication.

As it was concluded from the preliminary studies, students tend to design in many ways and methods, utilizing various types of software and media, both physical and digital. This was the basis for the decision made to not focus on designing inside the digital design studio. It was found to be most beneficial to let the students choose their own media for designing for the sake of supporting various needs and preferences, and focusing on other issues than producing CAD software. In combination with the support for various types of data, it was established that simple and fast pipelines should be developed in order to bring data into and out of the environment in a rapid way.

The decision to not simulate fabrication methods was made from a rather technical point of view in terms of the amount of specialized programming it would require. It should be stated that it could be beneficial to develop such features for a digital design studio, though looking from a technical point of view, the programming required to accurately simulate fabrication machinery and other more analog techniques fall into a very different category than the rest of the programming possibly required.

It was also decided that a game engine should be utilized to develop this digital design studio in order to have support for 3D environments and multi-user capabilities among other many useful opportunities that game engines bring. However, this will come with the cost of having to convert and export 3D models in mesh format, for being able to import them into the environment.

It should be stated that using a game engine to develop a digital design studio environment from scratch is only one possibility among many others. If given access to the source code, features could be added to any fairly sufficient existing software to better meet a design studio's needs. This could also be done to an extent through developing plugins and modding such existing software. Although, it should be safe to state that it can sometimes be much harder to modify existing software after a certain technical point, such as adding multi-user support.

Having chosen the development approach and reviewing the requirements, it was decided that humanoid user avatars in a 3D space should be used with this approach. This was mainly due to answering for the requirements of presence and scale perception. It should also be stated that choosing to use humanoid avatars comes with many implications such as camera modes, control schemes and movement capabilities.

4.2. Expansion

Keeping the chosen approach in mind, and the numerous research, software and video games discussed in the state-of-the-art review, a number of features were added to the list. These features are separate from the extracted requirements. Deriving from the possibilities in developing such an environment using the specified tools, these features should rather be considered as the advantages and the possibilities of a digital design studio. These features were first divided into general categories, and then explained in detail as individual topics as it is described below.

With the addition of the topics in this section, the resulting "feature list" is expected to partially answer the second research question (Q2), "For what purposes and to what extent can the potentials offered by virtual environments and game engine technologies be used in architecture, architectural education and distance education model?".

- **Analysis**

One of the most important advantages of being in a digital environment, or developing one, is the endless possibilities of analysis implementation and development. If certain analysis methods exist under code libraries or the utilized tools, they can be easily adopted. If the desired analysis methods do not exist, or are very specific to a certain condition or a developed feature, they can be scripted. Furthermore, anything that is being calculated in a manner can also be recorded and analyzed, such as user movements. It should be noted that for almost anything to happen in a digital environment, it must properly be defined, formulated and calculated whether in a higher or lower level, as in scripting or machine code. Shading, rendering, collisions, movement and even audio are being calculated in a level, and can be analyzed in such an environment. Below are just a fraction of the possibilities in analysis methods that design studios and architecture can benefit from.

Some analysis methods and subjects can also be simulated in addition to being calculated, which will be explained further under corresponding topics, in the category of "Simulation".

- Measuring Tools

Points in a digital environment are usually defined with either 2D or 3D coordinates depending on the environment. Even when arbitrarily placed, any point in a digital environment has its precise coordinates kept. Utilizing this common practice, distances between points can easily be calculated. Having these points keeping their X, Y and Z coordinates, it is also possible to only measure distances in certain axes, such as a vertical one. Being in a precise geometrical environment, surface normals, angles and many other calculations can easily be done, if not already being calculated. Some of these opportunities can be defined as: Measuring distances, elevation, angles between surfaces and calculating surface slopes.

Keeping in mind that game engines utilize mesh geometry which are composed of triangular and rectangular surfaces, which are called mesh faces, mesh geometry also keeps the face vertices, as in corners of these surfaces. Utilizing these vertices, various reference points can be taken from a geometry, such as snapping points for measuring tools' point placement. However in the case of custom/imported user objects, an uncontrolled amount of mesh faces and vertices may result in difficulty in such snapping capabilities, even though a geometry might seem like a planar surface, it may be composed of millions of smaller triangles.

- Shading

Shadows in a 3D virtual environment are often taken for granted. However, one important factor must be noted. Most shadows in video games are often baked, as in they are pre-calculated and merged with the texture of the surfaces they fall on, and commonly used rendering software take time to accurately represent shadows. This is mostly done to avoid computation costs and increase the performance of the software. In a digital design studio, it would be impossible to have baked lightmaps and shading as the models will be imported in runtime, thus not providing the advantage of being able to pre-calculate them. Due to this reason, real-time shading must be used at all times in a digital design studio. As Leitao et al. (2019) explain in their research, game engines are rather performative in such tasks and can produce sufficient results.

Moving on to the analysis opportunities in shading, having accurate shadows, in combination with accurate light sources can provide equally accurate analysis results. As it was mentioned in the previous topic, Mesh geometry is represented with mesh faces and vertices, utilizing these faces and vertices, lighting data on any of these can also be accessed. It should be noted that this data should be *accessed* and not *calculated*, on the account of it already being calculated by the graphics renderer, to be able to represent it on the screen. Any attempt to recalculate it will not only impact performance, but may also produce inaccurate results compared to what is being seen on the screen.

- Annual Solar Analysis

Either in combination with real-time shading or a cheaper algorithm in terms of computing costs, having an accurate sun path can be used to produce many analyses regarding the sun and shading.

Many sun path algorithms are present in almost every programming language and could easily be implemented. These algorithms often work with coordinate and date inputs, and provide calculated sun properties such as angle and distance as a result.

- Interior Lighting Analysis

Similar to the previous topics, through utilizing light sources and the geometry, interior lighting data, can be obtained. Through additional scripting, 2D daylight factor in compliance with common calculations can be made. Furthermore, as Bartosh and Philip (2019) have experimented with in their research, an equally spaced, 3D array of points can be used to visualize the lighting data in a volumetric manner.

- Structural Analysis

Although requiring a specific approach in importing models and an entire separate library of calculations, structural analysis can also be achieved and visualized. Having noted this possibility, it was decided that such analysis should be kept in specialized software due to the pipeline implications they bring. A simple model format, which is sought in this research, wouldn't contain any metadata regarding the structural properties of the components inside the model. Without such data, it would become very difficult to determine the structural integrity of elements and joints, and conduct such analyses.

- Space Syntax Analysis

Being an implementation of graph theory in architecture, spaces abstracted through the use of points and links can be utilized to easily calculate space syntax metrics. Many graph theory libraries and shortest path algorithm implementations exist for programming languages, and could be implemented into a 3D environment.

- Real Walking Distance

Another issue regarding shortest path algorithms, walking distances can be estimated through the use of such calculations. In combination with movement speeds and real distances, shortest path algorithms could be used to accurately represent the reachable points and areas, in terms of time or distance.

○ User Movement

Having actual users traverse in a 3D space represented with avatars that mimic human movement capabilities, their movement can also be recorded to analyze behavior. As such avatars can be abstracted with points moving through a space, on a programming level, the movement of these points can be recorded and visualized.

○ User Gaze

Tied to the previous topic and in combination with the recording of user movements, users' point of views can also be recorded to provide insight on visual focal points in a 3D space.

- **Simulation**

Similar to the possibilities in analysis methods, being in a digital, programmable environment brings the benefit of scripting simulations. Even though the digital design studio can be regarded as a simulation of a physical design studio, thus making everything a part of the simulation, the simulation methods stated here are aimed at pointing out the opportunities of programmable simulation methods in a digital environment.

○ Colliders

Perhaps not entirely a topic of simulation, but rather a pseudo-physical aspect in programming, colliders are one of the most important properties regarding a digital design studio. Simply put, colliders are the physical boundaries in a digital environment. A wall without a collider can be walked through, and stairs without colliders cannot be stepped on. Rendering and collisions are calculated separately, thus not everything visible can be bumped at, and a physical boundary doesn't need to be visible, e.g. invisible walls.

Colliders are most important in providing the means to simulate walking on objects and inside buildings. In combination with colliders, accurately represented humanoid avatars can help in investigating the ergonomics of certain openings and volumes inside a virtual space.

- Sun and Lighting

In addition to the methods discussed under analysis, light objects such as the sun or artificial light sources can also provide a simulation of lighting scenarios. When controlled through a sun path algorithm, a light source that imitates the sun can simulate accurate lighting and shading on objects.

- Ergonomics

Through utilizing procedural animations and inverse kinematics, character joints can be positioned through vectoral calculations. By doing so, certain movements such as walking on a slope, sitting on an object or reaching a specified point can be simulated on an avatar. When this avatar is observable, it could provide insight on the ergonomics of the associated design elements, thus such simulations come with the necessity of a third-person point of view.

- Accessibility

Discussed by several research, accessibility simulations can be achieved through altering colliders and character controllers. As an example, a wheelchair simulation can be developed where the surface slope is constantly being calculated to impact the movement, and a larger collider can be utilized to question the openings in the environment.

- Weather

Already utilized in most rendering engines, weather conditions can also be simulated visually. As an example: Through the use of surface normals, roofs and floors can be detected and filtered through raycasting upwards to see if they're outside and through the use of vertex displacement, snow accumulation can be simulated. Certain other elements in an environment, such as the skybox and camera filters could be adjusted to achieve the visual qualities of a specific weather condition. Either through visual effects or spawning small objects, rainfall and snowfall can also be simulated.

- Rain and Snow Accumulation

In addition to the visual aspects of weather, either through basic physics simulations or advanced fluid dynamics, rain and snow accumulation can be simulated.

- Wind

Similar to rain accumulation, but requiring advanced physics and fluid dynamics, wind simulations can also be achieved.

- 3D Sound

Commonly used in video games, realistic sound behavior can be simulated. Simple methods such as distance filtering or more advanced methods such as ray tracing and absorption can be simulated in a 3D environment. Through utilizing these methods or scripting common calculations, acoustics can also be analyzed through these simulations.

- Fatigue

Utilized commonly in video games as *stamina*, certain movements can be tracked to simulate fatigue when a predefined amount is surpassed. In combination with movement recording, shortest path algorithms and defining basic energy consumption per specific types of movement, estimations in calories can be made.

- Traffic

Employing the use of NPCs, drawing predetermined routes, or using shortest path algorithms with randomly selected points, pedestrian and vehicle traffic can be simulated. If developed further, moving into the field of game AI, NPC behavior can also be programmed.

- **Advanced Interaction**

Perhaps the line that separates a CAD software from a virtual environment, especially a multi-user one made with a game engine, is the range and capacity of interactions it should and could provide. The interactions between the user and the environment in a game-like space is often different than a CAD environment. This is mostly due to the way a user is represented in such environments and the associated control schemes. Below are some of the necessities and possibilities in using the chosen approach in this research. Although everything a user does can be regarded as interaction, the listed features are mainly focused on the methods and the media.

  - Object Spawning

Whether a user aims to measure distances or place prefab objects, "spawning" is required to achieve such tasks. The difference between imported objects and prefabs must be declared at this point. Imported objects are foreign to the environment and are not created by the developers, they can be attached with properties and scripts while importing, though such attributes can only be done through generalizing the properties they should have. This is due to the fact that the requirements associated with the imported objects can only be predicted so far. Some basic attributes that every imported object may need, such as computing and attaching colliders, calculating navigational data and shaders can be assigned while importing. However, more advanced scripting cannot be easily implemented on such objects.

On the other hand, prefabs are created by the developer and could be attached with advanced functions through scripting since they are persistent and not modifiable by the users. Distance measuring can be done through such prefabs, utilizing distance measuring objects, this task can be scripted in a way that each time two distance measuring objects are spawned, they shall calculate the distance between them and spawn another prefab that visualizes the distance in preferred format.

Aside from spawning scripted objects, and common in rendering engines, an object library can also be created by the developers to be utilized by the users. Furniture, light objects, roaming or static NPCs, environmental objects can all be spawned through a predefined object library.

- VR and AR

An advantage of developing a virtual environment with a game engine, VR and AR technologies can be utilized far beyond visualization capabilities. As an example, objects can be interacted with in a semi-realistic way through picking them up via controllers or spatial recognition through AR. All features and interactions can also be scripted particularly for these technologies, characters can be rigged to represent the users physical actions and be used to analyze ergonomics, furniture and building elements can be scripted to be more responsive.

- **Advanced Communication**

Most commonly used in 3D virtual spaces such as video games, there are many methods of advanced communication methods that can be developed further and implemented for a digital design studio.

- Spatial Communication

In a multi-user 3D virtual environment, the communication tools can also be developed to be both in 2D and 3D. 2D planes can be created inside a 3D space to provide surfaces for drawing or writing, and using the advantage of being in a 3D environment, these planes can also be used to superpose annotations to a specific, saved camera angle. These annotation tools can alo be taken to the third dimension through making them possible in 3D, as in 3D drawings or on-3D-point annotations.

- Animations

Aside from providing means for presence and ergonomics analysis, rigged and procedurally animated avatars can be utilized to simulate communication methods such as pointing with a finger, waving hands, nodding or shaking one's head. These animations can provide silent methods of communication and may be preferable in certain scenarios.

- Spatial Audio

Utilizing a 3D space can bring the opportunity to simulate realistic auditory communication. Audio sources can and listeners can be filtered on demand through distance and obstacles, simulating a physical environment. This could bring the advantage of being able to hold multiple conversations in the same environment.

- Communication Filtering

Similar to spatial audio, audio filtering methods can also be dependent on users. Private text messages or voice channels can be utilized to be able to hold conversations regardless of distance and without interruptions. It is worth noting that being in either a distance-filtered or user-filtered channel doesn't mean that your communication should only be limited to that filter. Commonly used in video games, users can be present in multiple channels and use separate triggers, such as keys or buttons, to submit their input to their channel of choice. As an example, one key could be mapped to make your voice be heard to all users, while another can make it only audible to those that are close to you.

- **Views, Camera and Rendering**

On the topic of visualization, several modes, methods and effects could be achieved. In a game engine, the user view is represented and rendered through a camera object. Game engines come with built-in support for rendering 2D and 3D environments, as well as several modes and scripting possibilities attached to them. There also exist several methods to produce visual effects and filters, with further possibilities which could be achieved via several methods of scripting and programming.

- Camera Modes and Properties

Orthographic and perspective camera modes are built-in in most engines, there are several adjustable properties associated with cameras that provide further customization options. One of these options is the ability to modify and script camera rotation and position. Through placing the camera inside a character or behind it, first person and third person perspectives can easily be achieved. Furthermore, input such as pointer movement can be converted into camera translation and rotation. Field of view and camera distances can also be modified in runtime to achieve zooming effects. Having these scripts and animations communicating with each other, it is also possible to translate the input regarding the camera into procedural animations such as head rotation.

On a more architectural level: Plans, elevations and sections can be achieved through utilizing said camera modes and properties in combination with shaders. It should be noted that having an orthographic camera slicing through an object is not enough to represent a section, the infill of this section relies on shader programming.

In addition to the aforementioned features, users have the opportunity to control their own cameras in a multi-user environment. Furthermore, these properties could also be communicated between users to have them share their views when necessary. This feature is often called a “spectator camera” in video games, where one user can opt to view the environment from another user’s perspective.

- Visual Effects

As it was stated before, not everything visible should be physical in a digital environment. Particle effects could be utilized to simulate certain particles such as rain and snow to achieve environmental effects on a visual level. Such effects could also be associated with certain interaction elements to provide increased awareness of actions, in terms of workspace awareness, consequential awareness and user experience. As an example, spawned objects could cast beams of light or have glowing particles around them to increase the awareness of another user's actions in addition to providing a more tactile feedback for the responsible user. These visual effects could also be combined with auditory feedback, as in sound effects.

- Shaders

Not only related to lights and shadows, shaders are an important part of visual programming in game engines. Shaders can be regarded as the final step in rendering pixels, a last step in representing the things that are happening in the digital environment. Being the final step, certain effects can be achieved outside of the actual geometrical state of the objects. As an example, visual properties of meshes can be programmed to achieve certain effects; programming regarding vertex displacement, glowing, transparency, infill and texture overlays fall into the category of shader programming. As an example, a sphere might not have strands of hair on it, but it can be rendered as a realistic fur ball with the help of a hair shader or a leafless tree model can have animated leaves through foliage shaders.

4.3. Technical Design

After reviewing the aforementioned requirements and possibilities, the technical design of a digital design studio was shaped. It should be stated that the technical design of a software is rather subjective and iterable in the development process. The chosen design is merely a starting point, and only one of the many approaches that could be taken.

First of all, a basic explanation of the expectations must be made. The aim is to produce an environment where users can bring in their work and communicate on it. The bringing of the materials should be as simple and fast as possible. Using the environment should also be as simple as possible, as it was discussed between the research supervisor and the author: “Something like a Zoom for architects”. The environment should support at least 20 simultaneous users to support a whole design studio group. The users should have the opportunity to have individual control of their perspectives on the material, and see the presence and actions of the other users. Annotation and communication tools such as sketching and writing text should be present. Regarding the material brought into the environment, certain analyses and simulations should be able to be produced. These analyses and simulations, in addition to the annotations, should be able to be brought outside of the environment. The environment should also provide the opportunity for the users to communicate with each other individually, as in having multiple conversations or interactions in the same space. Simply put, the aim is to develop a digital environment, mimicking a physical design studio.

As it was stated previously, due to being unable to modify existing software extensively, the approach of developing one from scratch was chosen. Having examined the existing research and software, it was decided that the platform for developing such software should be through utilizing a game engine. This decision was also made due to the fact that game engines offer many built-in features that are rather difficult to program from scratch, such as rendering, input handling, networking, physics and a system for combining them.

Having settled on the platform, the requirements, it was decided that the environment should be a 3D space. In this 3D space, certain 2D elements and material should also have the opportunity to be displayed on 2D planes, as in display boards and surfaces. One the other hand, the 3D elements should be able to be viewed in 2D, with the help of orthographic projection modes.

Starting with the users, as it was pointed out in the requirements regarding presence and workspace awareness, it was decided that they should be represented by humanoid avatars. These humanoid avatars will be rigged and animated to make the procedural animations and the other various animations possible. To provide a better sense of awareness, animation rigging concerning the spine and the head of these avatars will be done to reflect users' point of view. Furthermore, additional programming will be done to have these avatars pointing with their fingers on demand. The pointing will be done via raycasting to have the finger accurately point to the raycasted location from the center of the screen. This interaction will also be the trigger of various tools, pointing will have the camera zoom in on the pointed direction and enable the spawning of certain prefabs, such as the ones associated with measuring. Additional zooming will be able to be done while pointing with the mouse scroll wheel.

Avatars will have name tags above them to specify the users, and the users will have the opportunity to customize the appearance of their avatars to distinguish themselves. This will also help in cases where distant name tags are too far to be readable but the avatars are distinguishable due to their size. The size of the name tags could also be constant regardless of the distance, but this was found to clutter the screen space when there are many users.

Each user will only have control over their own avatars and will do so using the commonly implemented WASD control scheme. W, A, S and D keys will have the avatar and the attached camera move in the associated directions and the mouse movement will control the rotation of the camera. It was decided that the camera should enable for rapid switching between TPP and FPP, placing the camera either inside the head of the avatar or behind it. When the camera is behind the avatar, a displacement was found to be necessary to avoid blocking the user's vision. This will be achieved by the translation of the camera to either side of the shoulders, with an additional trigger to switch between the shoulders. Users will also be able to jump over small obstacles using the spacebar key.

Additionally, it was decided that there should be a no-clip flying mode, where the users will have the opportunity to fly through colliders and ascend or descend freely over, under and above all obstacles. The flying mode will also make it possible to have an aerial view of the environment.

Basic tools will be toggled from a toolbar, visible in the UI, and more advanced features such as simulation and analysis methods will be accessible through a context menu in the form of a window with tabs.

Avatar scaling was found to be an interesting opportunity in experiencing the environment and scale perception. Avatars will be able to scale themselves rather than the environment, as it will avoid complications in a multi-user scenario.

In the case of utilizing VR headsets, the users will still be represented by the same avatars but with a headset over their character's heads as it will specify which users are using VR. The control scheme will naturally be different in VR. In a room-scale scenario, users will have the opportunity to physically walk inside the environment. On the other hand, in a stationary scenario and for traveling further distances, teleporting through pointing will be implemented. VR users won't have an option for third person perspective to avoid motion sickness.

In either case, avatars will have procedural animations regarding their movement, point of view and foot placement. The animation rigging regarding point of views and finger pointing will be replaced with controller IK for VR users. Other animations such as waving and nodding will be done physically by VR users.

It is worth mentioning that the first scheme to be developed will be the mouse + keyboard controls. This is due to making the software accessible to more users, as it was determined that almost none of the students own VR headsets.

Users will have the opportunity to leave their avatars and have orthographic views of the environment or the material through employing tools associated with spawning cameras. Section planes will be developed with the help of shaders. It will also be possible to keep/save these cameras to be able to revisit them.

Both peer-to-peer connections and dedicated hosting will be developed for networking. Peer-to-peer will be aimed for private sessions between students, and dedicated hosting will be to provide a persistent digital studio environment, mimicking a physical one. The permanent sessions are expected to be hosted from dedicated servers at Istanbul Bilgi University. These permanent sessions will have de-escalated privileges for the students.

All interactions between the users and the environment will be in real-time, whatever a user does or observes will be reflected to others in an appropriate way, with the exception of certain cases such as, where the user may choose to temporarily have a different sun orientation or weather condition from the rest of the server.

In order to help with orientation, a compass and a minimap will be present in the HUD. The minimap will be centered on the user and have a top-down snapshot of the environment which will be updated regularly to accurately represent the changes. It will either be rotated with the camera orientation or have a representation of the view cone of the user. The minimap will also show the locations and the view cones of the other users.

Measuring and annotation tools will be networked prefabs, visible and editable by all users. Certain prefabs, such as distance measuring tools, will have the option to only be displayed for a duration while the others, such as sketches and notes will be permanent unless deleted.

Text and voice communications will be implemented. These communication methods will have distance and channel filters. A video communication method was also suggested, where the users can have their camera streams projected on top of their avatars. However, this will significantly increase the network load.

A spectator mode for the camera, where the users can observe other users' cameras will also be developed. Additional interaction methods for the spectator mode were also found to be necessary, such as directly drawing on a user's view. Another form of spectating, named "observing" was also found to be necessary. In the observer mode, users will either be using previously placed cameras or have a separate, presenter controlled camera to observe the environment, rather than directly watching other users' cameras. This was found necessary to help with the users unfamiliar with the environment and the WASD scheme, e.g. guests.

Some of the analysis and simulation methods will be done on the users' side and not visible to others to reduce computation costs, such as space syntax analysis. The rest will happen simultaneously on all users with the option to opt out, such as sun orientation.

Logs regarding user participation, movement, gaze and communication will be recorded by the server to let some analysis to be conducted and to be exported on demand.

Importing will happen at runtime and in session. This will bring the opportunity to communicate on multiple subjects in a single session, either simultaneously or consecutively. This way, it will also be possible to update the imported objects without restarting the session. This was found most necessary for permanent sessions. However due to this approach, any imported objects will have to be tracked and reflected to each user, including late-joins. This will result in heavy server loads and might better be kept on client-side with a separate solution for sharing models. If this route is taken, the imported objects will be asynchronous, meaning there will have to be another pipeline regarding sharing these models and each user will be responsible for what they have locally, implying that the session server will only tell what file to use, not give it to the user directly. This can be achieved through utilizing a separate FTP server or leaving responsibility to the users. When manipulated, this can result in users seeing different objects in the same session, although countermeasures can be taken. In addition to the advantages on server loads, this will also let users have their own pace while downloading models, rather than the server keeping busy with each user and disconnecting them from the session due to timeouts or connection problems.

3D models will have to be attached with colliders during importing. They will also be tagged to keep track of them in order to associate spawned objects with the models they are spawned on. This will bring the opportunity to export some of the annotations and prefabs back to the CAD environment.

Regarding the user material to be imported, the 3D models will have to be in mesh geometry and a generic file format such as FBX or OBJ. This will require a pipeline either through a scripted exporter for CAD software or through informing the users. The images will also have to be in certain supported formats such as PNG and JPG. Working with mesh geometry demands proper modeling etiquette, as in paying attention to face orientations and mesh quality.

Exporting will only happen on the client-side. Users who wish to download analysis data or session logs will have the opportunity to do so on demand. Not every interaction will be able to be exported and some data will have to remain in the environment. Export data will be in common CAD or text format, such as DWG, FBX, OBJ, JSON and TXT.

It was determined that any further imports, exports and live interchanges between the environment and a CAD software will need to have corresponding import/export plugins developed.

4.4. The Feature List

Following the technical design, a table to keep track of, update and assign priorities to planned features was made. This feature list will be updated regularly during development in the prototype-test-feedback loops. The categories, features and short descriptions of the entries from *the feature list* are cataloged below. In the *feature list* used in the development of the software, these entries are tracked in priority and status, and small notes and pseudocodes are noted next to them. In addition to *the feature list*, Git (URL-25) was used for version control and tracking the changes.

Table 4.1. “Character” category of the feature list

Category	Feature	Sub-Feature	Description
Character			Player character, the avatar that represents the user
	Humanoid Avatar		Bipedal humanoid avatar to be controlled, rigged and animated
		Avatar Picker	Ability to choose from a range of avatars
	Basic Locomotion		Basic character movement
		4 Axis Move	4 Axis character movement
		Run	Increase speed on sprint prompt
	Jump		Basic character jump
		Gravity	Realistic gravity simulation with fall acceleration and terminal velocity
	Fly		Basic character flight
		No-clip	Flight without clipping (ghost mode) for both the character and the camera
	Teleportation		Point and teleport for VR
	Size Manipulation		User self size manipulation
		Fixed Scale	Predetermined scale set on prefabs, activated on collision
		Free Scale	Adjustable character scale

From a programming point of view, the features associated with the character also include the scripts directly attached to the character model. Even though the character animations and the camera are also attached to the character, they are quite distinct from the code regarding the character itself.

Table 4.2. "Character Animations" category of the feature list

Category	Feature	Sub-Feature	Description
Character Animations			Character model animations
	Locomotion animations		Realistic character animations based on movement type and direction
	Foot IK		Inverse kinematics on foot placement (realistic foot placement on surfaces)
	Head rig (look)		Character head rotation based on camera direction
	Hand rig (point)		Finger pointing on prompt
	Dynamic animations		Setup for animations which play on top of movement or other animations
		Hand Waving	Waving animation, "Hello!"
	Static animations		Setup for static animations that stop other states
		Sitting	Cross-legged sitting
	Procedural animations		Setup for animations that utilize IK to locate bones
		Butt IK	For testing seating surface ergonomy
		Arm's Reach	For testing arm's reach in certain spaces, e.g. bathroom sink
		Body Reach	For testing body reach in certain spaces, e.g. kitchen cabinets
	Looped animations		Setup for animations that loop unless prompted to exit
		Dancing	Seamlessly looped dancing
	Animation rigging		Setup for blending animations and IK on top of each other, e.g. head rig while sitting

Certain animation modes require them to be set up in scripting and animation programming beforehand, after this initial programming, more animations of the same type can be added easily.

Table 4.3. “Camera” category of the feature list

Category	Feature	Sub-Feature	Description
Camera			User camera, screen space renderer
	TPP		Third person perspective, over the shoulder follow camera
		Shoulder Swap	Shoulder swap with hand swap for hand rig
		Perlin Noise	Small perlin noise to mimic breathing
		Mouse Smoothing	Input smoothing to reduce jittery camera movement in TPP
	FPP		First person perspective, raw rotation through mouse input, disabled effects
	Camera Switch		TPP/FPP Switching
	VR		Stereoscopic rendering for VR headsets
	AR		Semi-AR utilizing VR passthrough
	TPP Zoom		Adjustable character distance from camera on TPP
	Pointing Zoom		Zooming on either perspective modes
		Distance zoom	Decreasing/Increasing distance to object
		FOV zoom	Decreasing/Increasing field of view to achieve zoom effect
	Camera collision		Camera collider to avoid clipping
	Orthographic Projection		Orthographic/Isometric camera mode
	Spectator Camera		Viewing from another user's camera
	Observer Camera		Presenter controlled camera
	Camera Saving		Saved camera transform (position and rotation)
	Shaders		Shaders that are required to achieve certain view modes
		Section Plane	Section plane with section infill shader
		X-Ray	See through, wireframe shader
		Hard Edge detection	Shader highlighting hard edges
		Arctic Mode	Mono-material shader, e.g. plain white

Table 4.4. “User Interface” category of the feature list

Category	Feature	Sub-Feature	Description
UI			User interface, screen space overlay
	Main Menu		Landing/Launch menu
	Connection Menu		Online connection menu
	Pause Menu		In-session pause menu
	Help Menu		Help menu visualizing controls and hotkeys
	User HUD		User heads-up display
		Compass	Compass HUD
		Minimap	Basic minimap of the environment
		Crosshair	Screen center indicator, reticle. Useful for pointing and spawning
		Slope text	Current position slope indicator text
		Toolbar	Toolbar HUD, for displaying tool and menu hotkeys, and the active tool
		Chatbox	Text chat box
	Interaction Menus		Context menu holding tools outside of the toolbar
		Environment Tab	Tab containing environmental controls, such as the sun and the weather
		Analysis Tab	Tab containing analysis tools
		Simulation Tab	Tab containing simulation tools
		Character Tab	Tab containing character options, such as the avatar picker
		Prefab Tab	Tab containing spawnable prefabs, such as furniture and trees
		Import/Export Tab	Tab containing import/export options

As it was previously mentioned, not all tools can be mapped to hotkeys or a toolbar. In order to reduce cluttering on the screen and provide a better user experience, most tools aside from the very basic, most frequently used ones were moved into a context menu that is accessible with a hotkey.

Table 4.5. “Analysis” category of the feature list

Category	Feature	Sub-Feature	Description
Analysis			Analysis tools, mostly concerning the imported 3D objects
	Slope Angle		Slope angle in percent for the point that the character is standing
	Shading		Real-time; point, area or surface based lighting amount
	Solar		Annual solar analysis, through imported library
	Daylight Factor		Real-time interior lighting analysis
	Space Syntax		Space syntax analysis through abstraction of the space i.e. nodes and links, through imported library
	Walking distance		Walking distance visualization by real footstep count, outward navmesh
	User Log		Recording user actions
		Participation	Timestamped connection, disconnection log
		Movement	3D movement log
		Gaze Heatmap	Conic raycast log, with intervals

During the phase of creating a feature list from the previous work, some features such as structural analysis were found to be too difficult to implement regarding the other features and the importing pipelines. The entries in the feature list were either added through previous experience or were questioned in logic through pseudocoding. Whenever a feature was found to be incompatible with the rest of the systems, or too difficult to implement, it was discarded.

Table 4.6. "Simulation" category of the feature list

Category	Feature	Sub-Feature	Description
Simulation			Simulation tools, mostly concerning the environment
	Basic Sun		Basic 2-axis sun rotation
	Sun Path		Location, date and time based realistic sun path, through imported library
	Wheelchair		Wheelchair accessibility simulation
		Wheelchair Constraints	Physical constraints: Slope limit, step height and larger collider
		Wheelchair IK	Wheelchair and player model rotation according to current point normal
	Weather		Visual weather effects
		Skybox	Separate skyboxes to accompany weathers
		Sun Properties	Separate sun attributes to match skyboxes and the weather
		Rain and snow	Particle effects or screen space 2D overlay to simulate rain and snow
		Wind	Foliage shader affecting wind property, only possible with object library trees and foliage
	3D Sound		Raycasted sound, affected by distance and obstacles
		Sound Objects	Spawnable, seamless looping audio sources for audio mapping
	Fatigue		Fatigue log through step count, paying attention to slope and vertical movement. Could be calculated in calories.
	Driveable Vehicles		Driveable, friction based vehicles for realistic parking/street maneuver analysis
	Pedestrians		Navmesh and pathfinding controlled NPCs
		Spawn Frequency	How often peds spawn
		Density	How many peds spawn on a single spawn instance
		Spawn Points	Where they spawn, could be random
		Target Points	Where they move towards, could be random
	Rainfall and Snowfall		Rain and snow accumulation
		Basic physics objects	Rain and snow balls that collide with the environment, bounce and accumulate according to slope
		Fluid dynamics	Realistic liquid accumulation, separate behavior for snow and rain
	Wind		Wind fluid dynamics

Table 4.7. “User Interaction” category of the feature list

	Wind		Wind fluid dynamics
User Interaction			User tools, spawnable prefabs
	Temp Tag		Temporary tags, for spontaneous and precise pointing
	Perm Tag		Permanent tags, can store content
		Text store	Storing text messages inside perm tags
		Image store	Importing images into perm tags, such as details
		Voice store	Storing voice recordings inside perm tags
	Distance Tool		2-point distance measuring tool
	Elevation Tool		Elevation tags, relative to the first placed tag
	Slope Tool		Point slope tag
	Angle Tool		3 point - 2 line angle measure
	Object Spawning		Spawnable objects
		Object Library	Premade basic static objects such as furniture and trees
		2D User library	User-uploaded 2D objects to be placed on display boards, surfaces or planes for superposing
		3D User library	User-uploaded mass-spawnable 3D objects, smaller than architectural models
	Display Boards		Interactive 2D surface prefab for 2D assets
		2D Asset Placement	Content picker interaction
		Pan and Zoom Camera	Orthographic camera with pan and zoom controls, perpendicular to the board
		Asset Layering	User layers on base assets (e.g. analysis layers and pages on top of a base map)
	Camera Saving		Camera saving interaction
		Asset Hiding	Making certain objects visible only in specific saved cameras
	Destination Marker		Points to be set by users to be indexed and later teleported
	Light		User-spawned light sources
	Paint		Painting tool
		Spray Paint	Paints on top of surfaces with a radius, separate layer
		Paint Bucket	Paints whole surface/object/prefab, material overlay
	Air Sketch		Superposed 2D sketch on a 3D static plane
		Free Sketch	Free hand sketch on plane
		Line	Points to lines
		Circle	Center point and radius to draw circle
		Stored Camera	Store superposed perspective cam to sketch object, hide drawing inside
	3D Drawing		Drawing in 3D
		3D Lines	3D points to lines
		Spheres	Center point and radius to draw sphere
		Polysurfaces	3D points to vertices, to triangles, to 3D polysurfaces

Table 4.8. “Communication, AI and Networking” categories of the feature list

Category	Feature	Sub-Feature	Description
Communication			Communication methods between users
	User Name Tags		Over-the-head name tags on character models
	Text Chat		User text chat box
		Private Messaging	Private text messaging between users
		Channels	Separate channels for separate groups or purposes
	Voice Chat		User voice chat
		Global	Global voice chat (where everyone can hear each other)
		Local 3D	Local 3D voice chat (Distance filtered, realistic)
		Chat Rooms/Groups	User/Group filtered voice chat
	Video Chat		User camera projection on alternative user model or a surface
	User screen space overlay		Screen space overlay for drawing and annotations, for spectators and observers
AI			NPC AI capabilities
	Pathfinding		Obstacle avoiding, shortest path utilization for NPC agents
		Origin to Target	Individual origins and targets set by the user, or randomized within rules
		Evacuation	Single target for all present NPCs
Networking			Networking schemes and methods
	P2P Hosting		Peer to peer hosting between users
		Relay	Third-party relay to increase security and decrease operations regarding hosting
	Dedicated Hosting		Dedicated hosting from a non-user server
		Privilege De-escalation	Less control for connected users other than an appointed admin, to avoid flooding
	Connection Approval		Approval logic for incoming connection
		Password Protection	Password protection for sessions

Table 4.9. “Input/Output and Scene” categories of the feature list

Category	Feature	Sub-Feature	Description
I/O			Input/Output methods and options
	User Objects		Objects that belong to the users, that are initially stored on their end
		Static import	Importing user objects in editor, built-in objects for special cases
		Realtime model import	Importing 3D user objects in runtime
		Refresh import	Check original file, refresh import on change
		2D Asset import	Importing 2D user objects in runtime
		Privacy Filter	Filtering whether an object is visible to other users, with variable duration. Useful for permanent sessions.
	Session Objects		Objects that are instantiated in runtime
		Tag export	Exporting placed tags (position)
		Placed object export	Exporting placed objects in place (position and rotation), such as objects from the object library
	Session Logs		Session recordings regarding users
		Chat	Text chat log (.txt)
		Movement	User movement log (x,y,z .json)
		Gaze heatmap	User raycast log (x,y,z .json)
	Rhino Model Exporter		Rhino exporter plugin for exporting models in proper settings and format
Scene			Features related to scenes
	Pre-Made Lobbies		Lobbies to be used in permanent sessions
		Classroom	Defined areas for each group/student
		Presentation room	Bigger scale display boards, smaller model display stand for microsizing
	Scene Layers		Layers for keeping objects, helps with exporting
		Spawned prefabs	Tags etc.
		Imported objects	User objects
		Scene modifications	Recorded, revertible modifications

5. DIGITAL DESIGN STUDIO CASE STUDY

Following the technical design and the preparation of *the feature list*, the development of a prototype was initiated. Unity was chosen as the engine, and some prototypes developed prior to this research were combined and used as a template for the software to be developed.

The utilized template was conceived mainly through merging two projects, titled “FPS Prototype” and “TPS Prototype”. Both of these prototypes had player avatars and animations taken from Unity’s first-person and third-person templates which were released around the time development began. Scratching the built-in programming of these templates, movement and camera programming corresponding to each of the perspective modes were reprogrammed on a basic level, with prearranged room for improvement.

Through merging these two projects, a template which offered both perspective modes was developed. Furthermore, another template titled “Netcode Boilerplate”, which was done to provide a simple starting point for any multi-user project utilizing Unity’s *Netcode for GameObjects* (URL-26) was used to add multi-user capabilities.

The aforementioned prototypes and templates were developed by the author as solo development projects, which was also the case for the rest of the software development made in parallel to this research. The development itself, regarding the programming/coding done, is outside of the scope of this research and was carried out voluntarily in parallel to the research.

The rest of the programming was mainly focused on adding the features in *the feature list* to this foundational prototype.

The prototype was eventually named “Online Virtual Studio” and was referred to as “OVS”. OVS was tested continuously during its development, in the scope of this research.

5.1. Second Design Studio Tutor Interview

After certain features such as annotation objects and static importing were developed for OVS, a second interview with the design studio tutors was held. Multi-user capabilities were not introduced yet. Therefore, the prototype was showcased through screen sharing in a video conference session. The initial showcase and the following discussions were conducted on 17/09/2021, and lasted 71 minutes.

At this state, OVS had the following features:

- **Humanoid Avatar**

A user controlled, humanoid avatar was present. The avatar had full body rigging and locomotion animations, with the addition of head rotation and finger pointing done with IK. Furthermore, foot IK was also implemented, the avatars legs and feet would react to the surface(s) it stood on, through position and rotation. The control scheme for the avatar movement was chosen as the commonly used WASD + Mouse scheme. Aside from the 2-axis movement, the avatar was able to jump and fly.

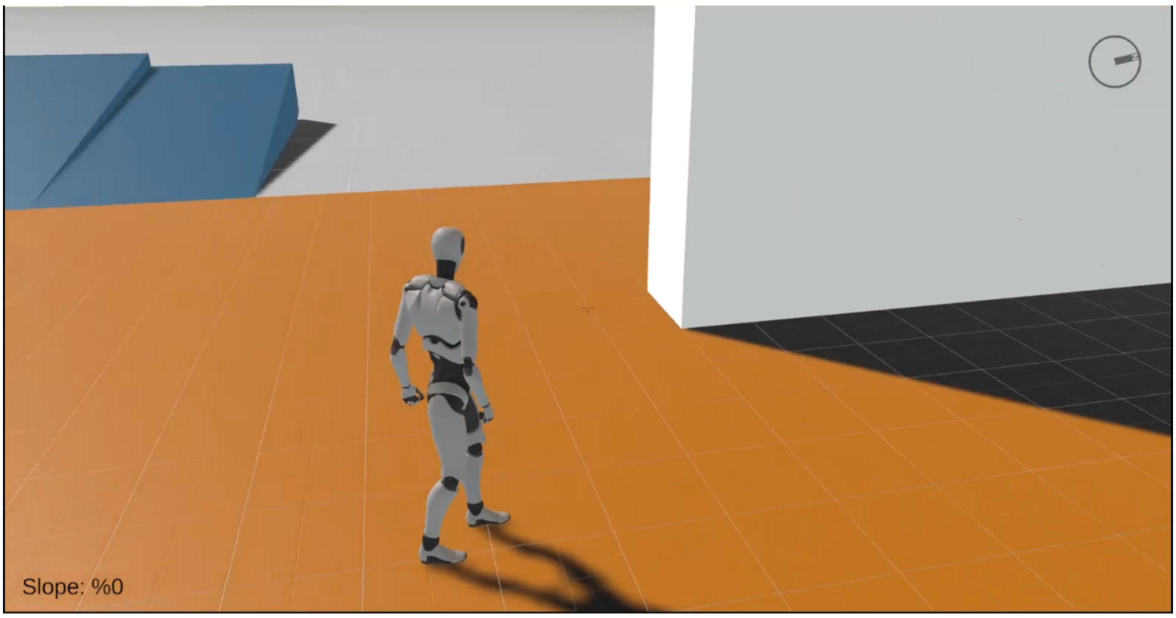

Figure 5.1. Humanoid Avatar

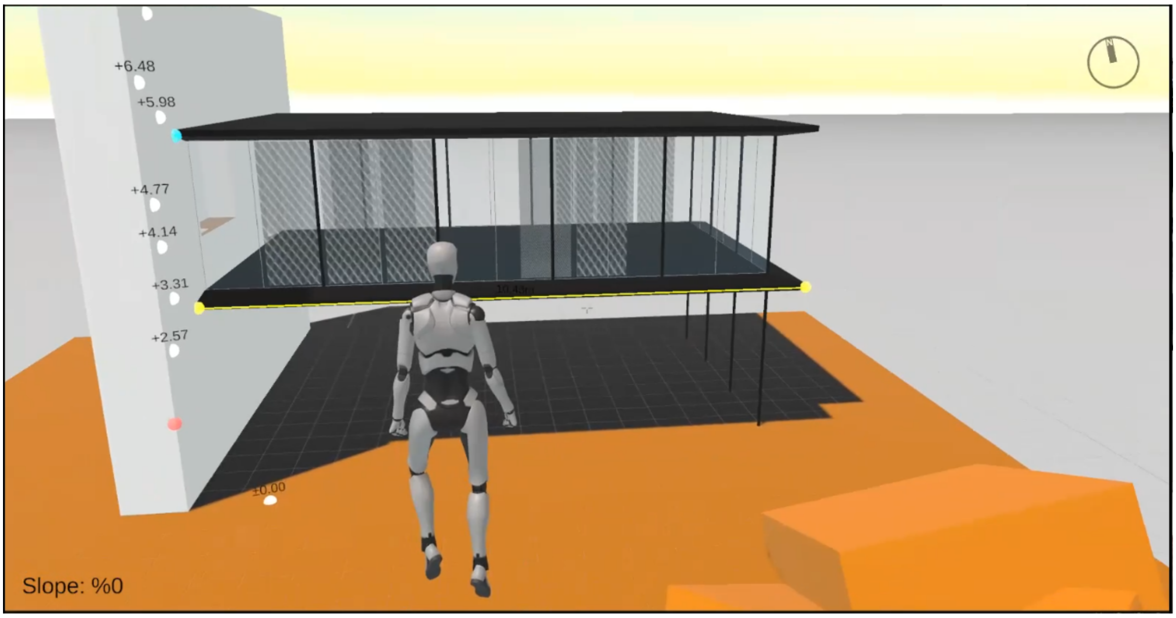

Figure 5.2. Flying Avatar

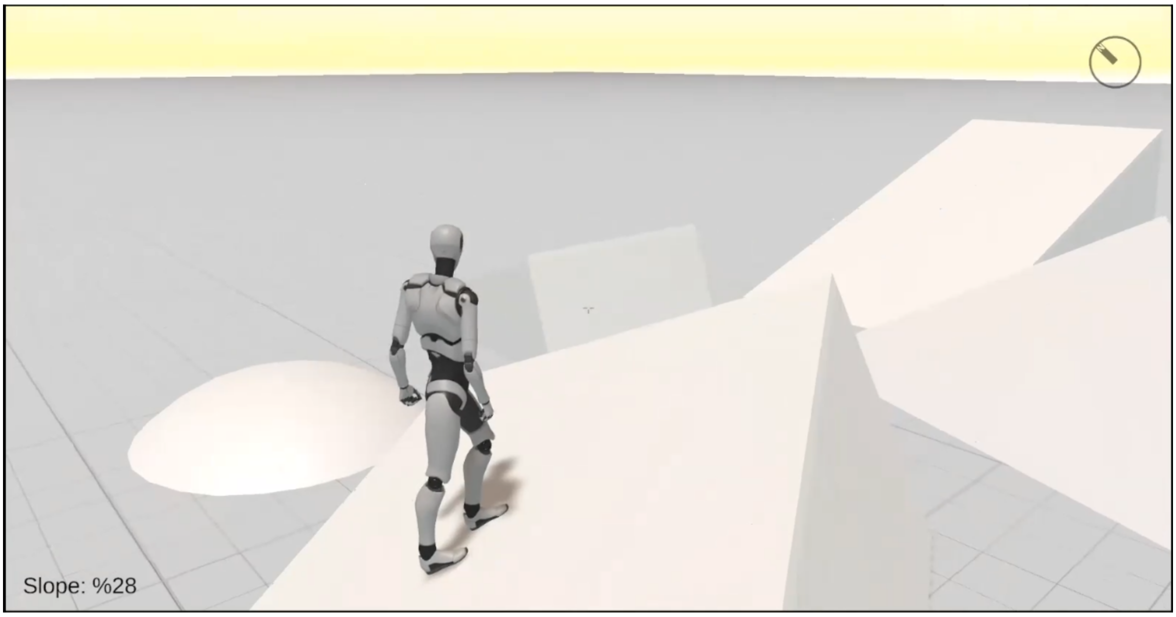

Figure 5.3. Foot IK

- **Avatar Scaling**

As it was discussed within the technical design, rather than scaling the models, it was decided that the avatars should be scaled. This feature was one of the first ones to be implemented.

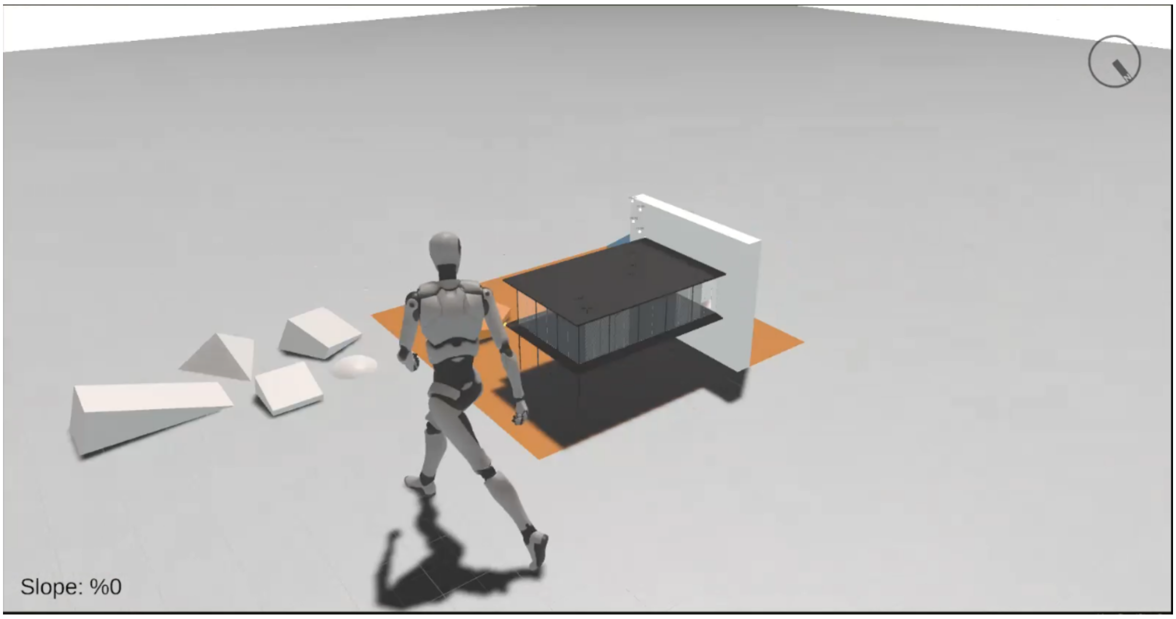

Figure 5.4. Super-Sized Avatar

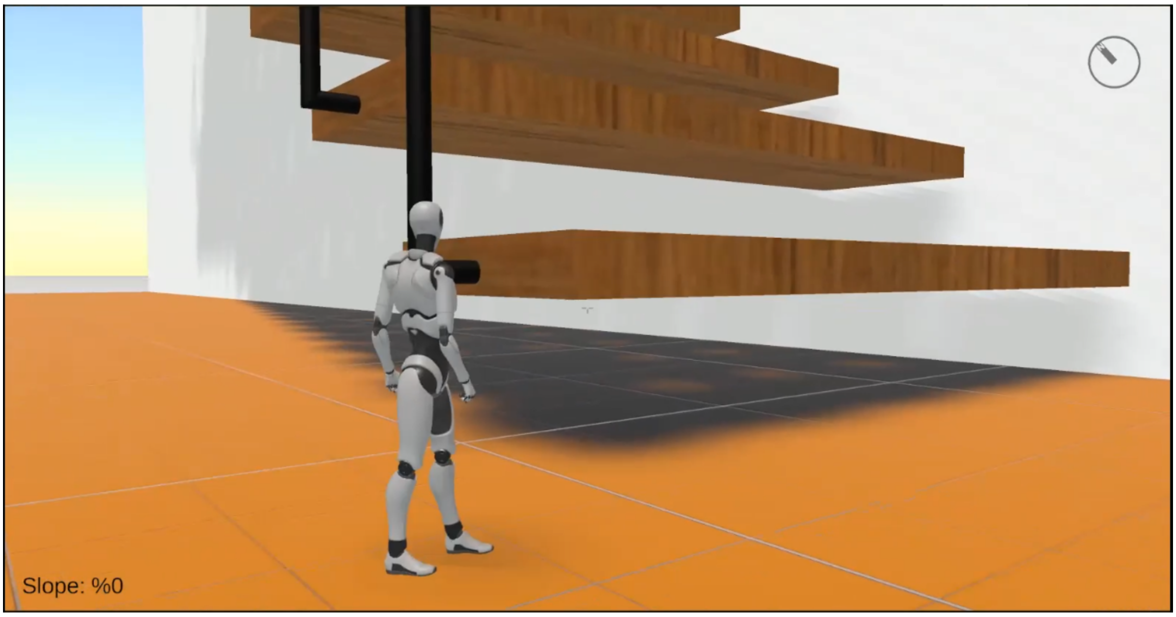

Figure 5.5. Micro-Sized Avatar

- **Camera**

An over-the-shoulder, shoulder-swappable TPP and a through-the-eye FPP camera was implemented. A switch between the two was done through the F key. The avatar's body movement followed the look direction through the IK rigging in both modes.

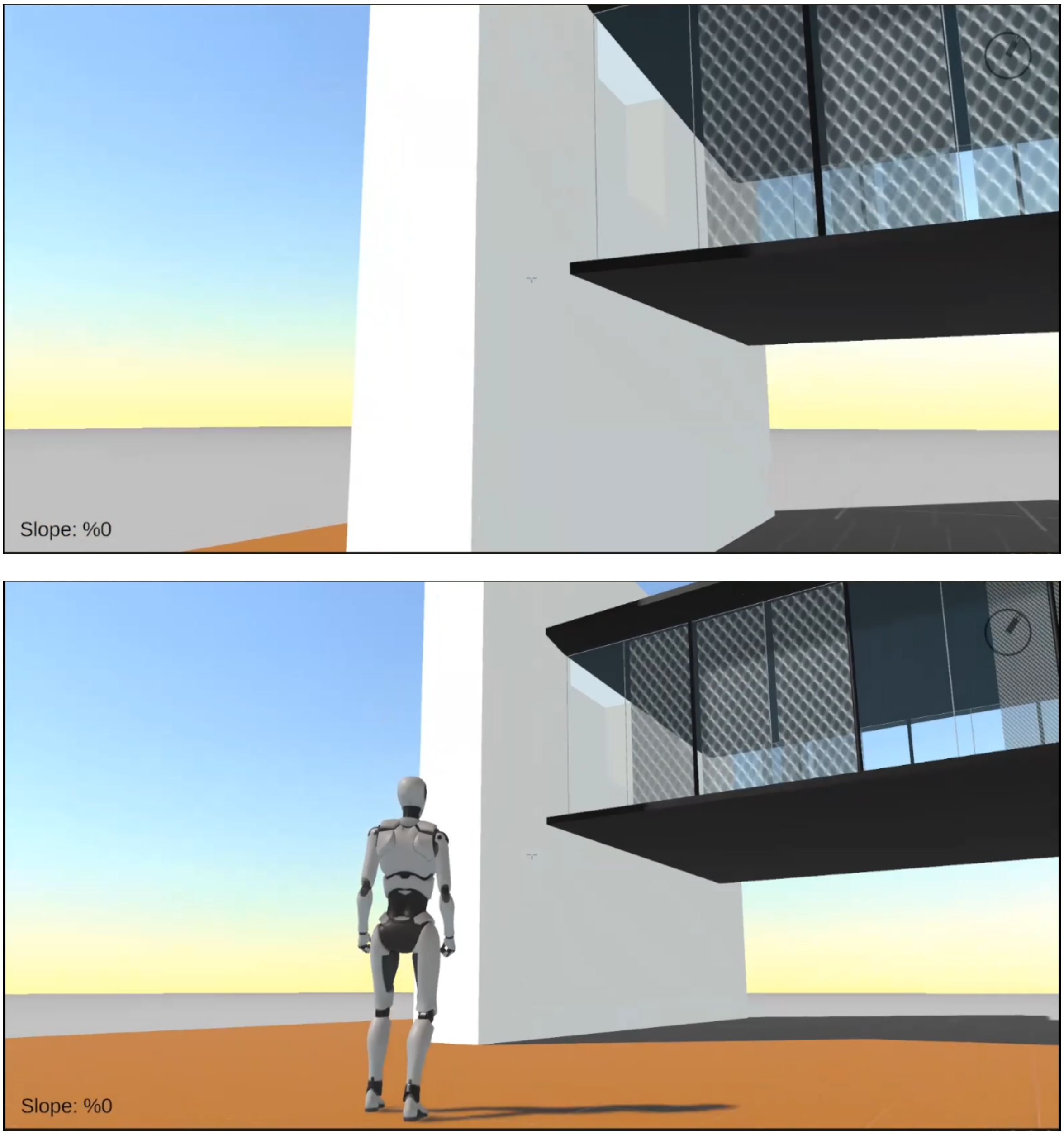

Figure 5.6. (a) First-Person and (b) Third-Person Perspectives of the same view

- **Annotation Objects**

Through the pointing interaction, annotation objects were able to be spawned. The available annotation markers were: A temporary spherical tag which moved when respawned, permanent spherical tags with another trigger to delete them, distance measuring tags, and elevation tags.

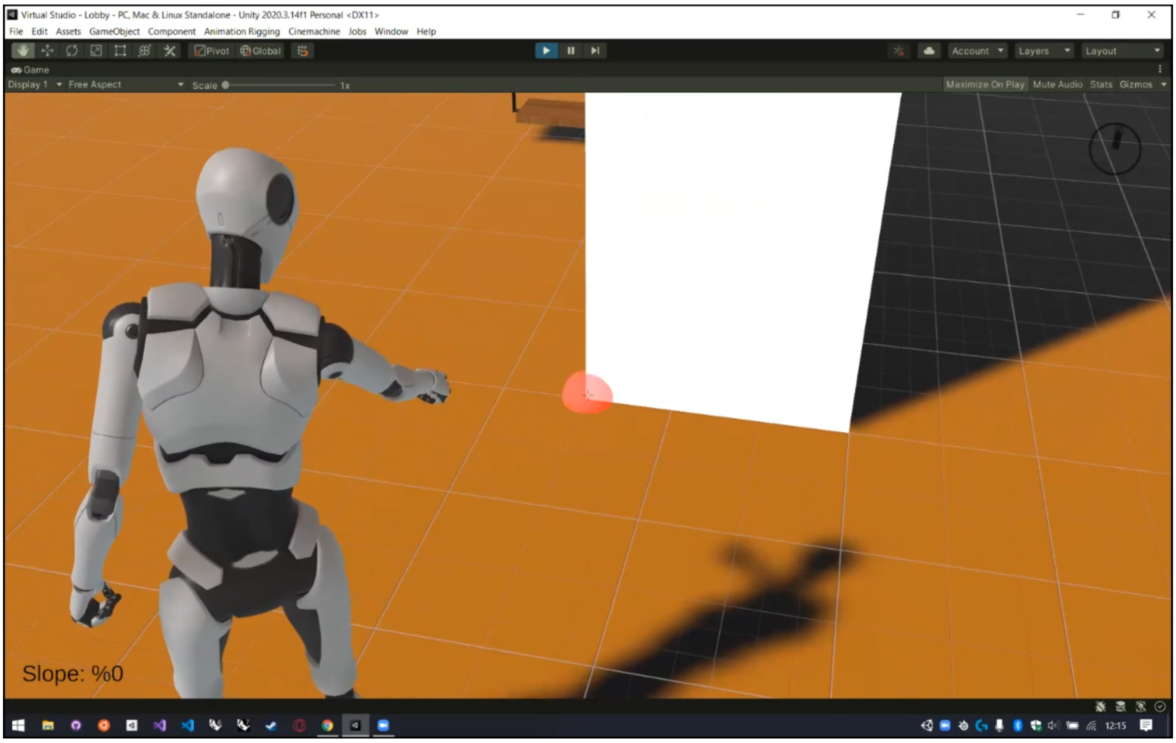

Figure 5.7. Pointing and spawning a temporary tag

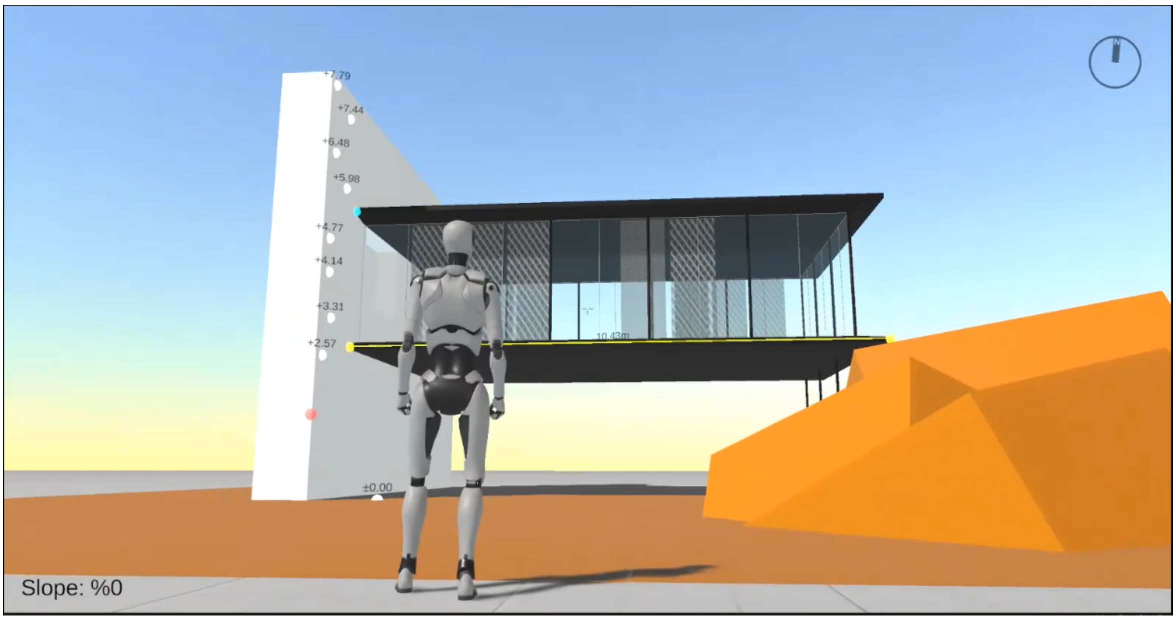

Figure 5.8. A model with several annotations

- **Environmental Controls**

A 2-axis sun control menu was implemented with the shading of the imported object being calculated in real-time.

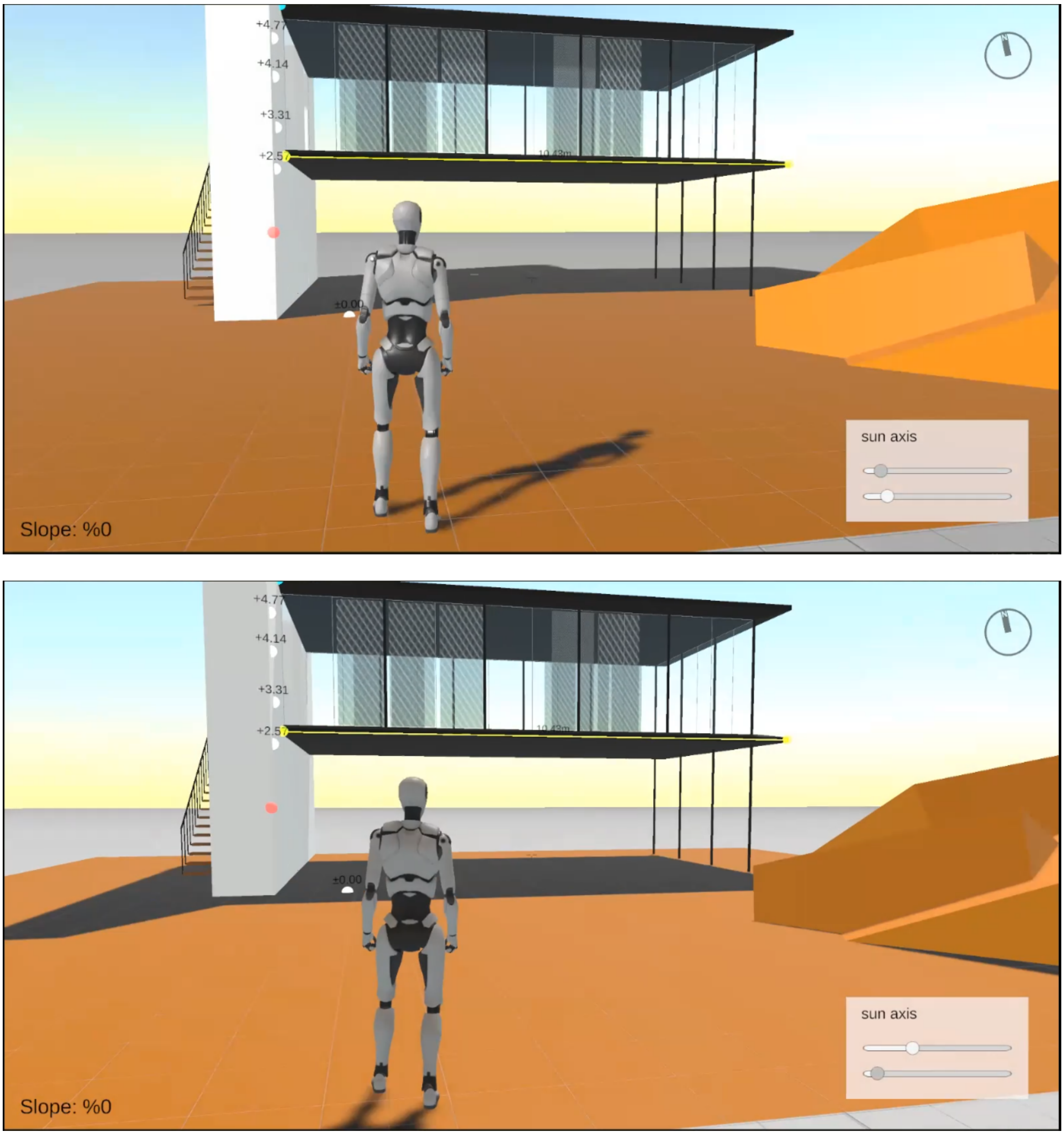

Figure 5.9. Controlling the Sun

- **Analysis**

A basic compass and a slope HUD was developed. The slope HUD could calculate the slope of the point the avatar stood on, in percent.

- **Simulation**

A wheelchair simulation was implemented. Utilizing the character controller programming, the avatar sat on a wheelchair and its movement was restricted. In wheelchair mode, the avatar would move %5 of its walking speed while climbing slopes higher than %10 to indicate that it's not suitable for accessibility, and it couldn't take a step larger than 5 centimeters in height.

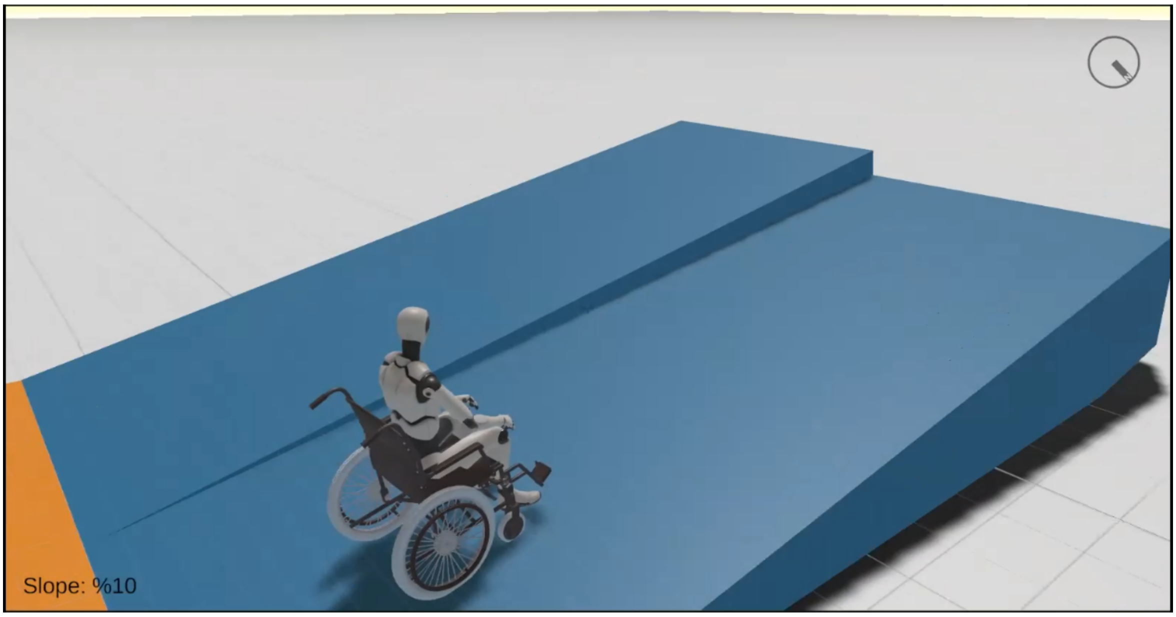

Figure 5.10. Wheelchair Simulation

In addition, a basic rainwater accumulation simulation was developed. Utilizing small bouncy spheres with colliders, which are also affected by gravity, a very basic simulation of rainwater accumulation could be made.

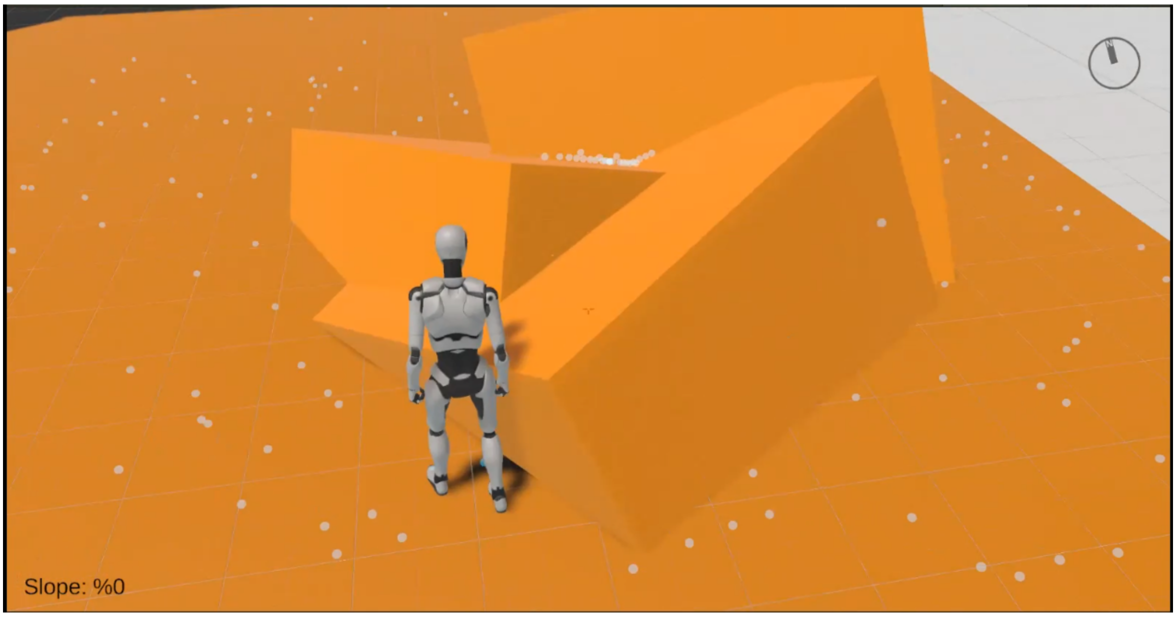

Figure 5.11. Rainwater Accumulation

The most important aspect of this interview was to determine the priority of the planned features and the relevance of the implemented ones. As this research was conducted in a limited time and the software development was done individually, it was accepted that only a small portion of the features might have a chance to be developed and tested.

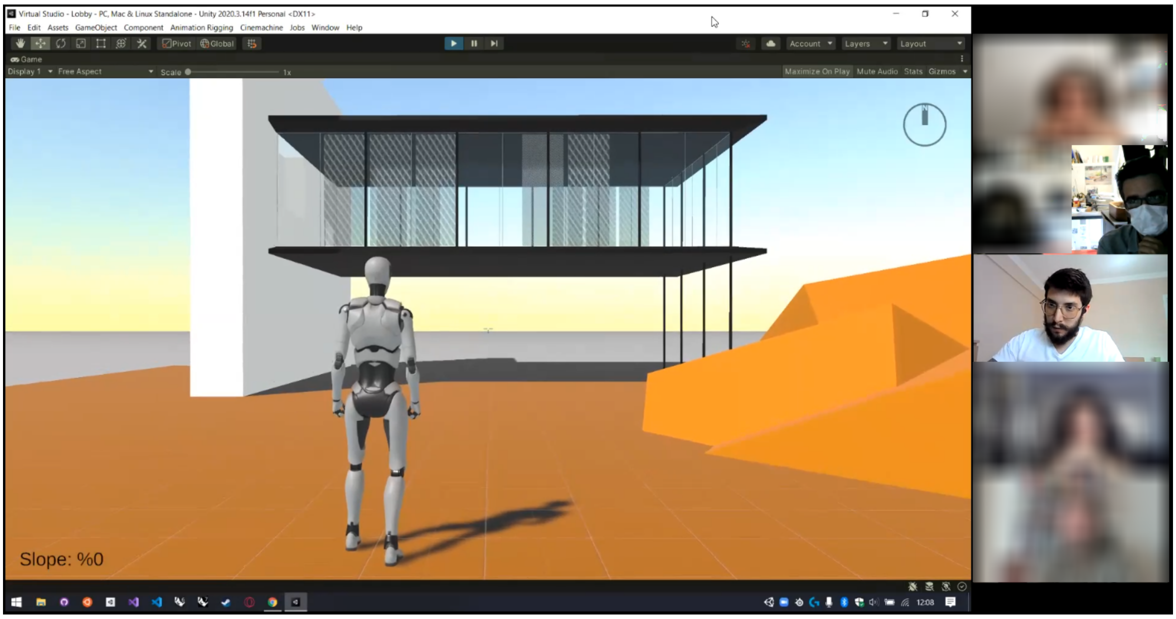

Figure 5.12. OVS Demonstration Video Conference

First of all, the scope of the research and the aim for developing such software was explained. Then, the previous research in this field and the findings of the preliminary studies were summarized to create a frame for discussion. After the initial explainings were done, the existing features and capabilities of OVS were demonstrated to the tutors, and their comments were collected.

The first comments received were on orthographic projections. Tutors stated that they would prefer to communicate on conventional plans and sections, rather than a 3D model.

However, they found the use of avatars to be educational. They explained that having a realistically proportionate avatar attached with a collider, which can bump their head to a poorly designed staircase and get stuck in small openings would point out such poor design choices. They added by saying that a human-sized avatar would help students grasp the scale of their design more easily.

On another topic, tutors suggested that such an environment should also let some alterations be made on the model, even if it’s not going to be attached with tools regarding CAD. This was answered by the author through explaining the intention of providing a fast pipeline between CAD and OVS. By utilizing such a system, users would be able to do the modifications in their preferred CAD environments and bring the changes into OVS in a couple of seconds. This discussion was carried further with the tutors suggesting that at least a drawing board or wireframe modeling should be present to more efficiently communicate and shorten the feedback loops through rapid sketching. Unbeknownst to them, this was a planned feature coupled with camera superpositioning and it was already in development, which will later be called “Airsketching”.

The thought of having a permanent digital studio, especially outside of class hours, was found to be very positive for the students. It was discussed that this would enable students to explore each other's works and perhaps leave notes on them, expressing some ideas. It was also stated that this would simulate a learning environment much like a physical studio space where students spend most of their time.

Regarding analysis and simulation methods, the tutors suggested that they shouldn't be prioritized as the students tend to do them on a very basic level if they do at all. Adding that such analysis can be done in more professionally developed, specialized software and perhaps should be left aside in terms of more efficiently using the limited time for the research. On the other hand, the already existing analysis and simulation methods were found useful for rapidly questioning some qualities on a basic level.

Even though multi-user capabilities or communication tools didn't exist at the time, when the character animations such as hand waving and sitting were shown, the tutors responded by saying that such animations could also be recorded somewhere in case anyone missed a reaction. Further on the topic of communication, tutors stated that spatial audio and communication channels would be most beneficial, as they found the lack of the possibility to hold multiple conversations in an online session was one of the biggest difficulties of the distance education period.

As a criticism for the whole environment, tutors explained that producing 3D models without attention to detail and materials, and rendering them in a realistic or a semi-realistic engine often leads to unsatisfactory results. They added by saying that using a 3D environment for communication might influence such outcomes. Tutors described that in the early stages of design, students often tend to do massing models and these wouldn't be aesthetically pleasing in this environment and might have some negative impact on the perception of their work. They suggested that simpler view modes and shaders, such as a wireframe mode and arctic shader, could be incorporated for these earlier stages.

5.2. Third Design Studio Tutor Interview

Following the previous interview certain requested features, as well as planned ones, were developed. A third interview was done with the design studio tutors to discuss the recent developments. The most significant development was the ability to hold multi-user sessions. Tutors were invited to join the session, using OVS. Due to the communication tools not being developed yet, a simultaneous video conference was held to provide auditory communication. The live demonstration and the discussions were done on 03/03/2021, and lasted 85 minutes.

The following features were added between these two interviews:

- **Multi-User Capabilities**

Client-server multi-user capabilities were developed. One user being the host, the other users were able to join the session through the host's IP address. The users could input their names which would be displayed on top of their avatars. Every action of the avatars were being reflected to each user simultaneously. Due to an analysis on the server loads, Foot IK synchronization was disabled.

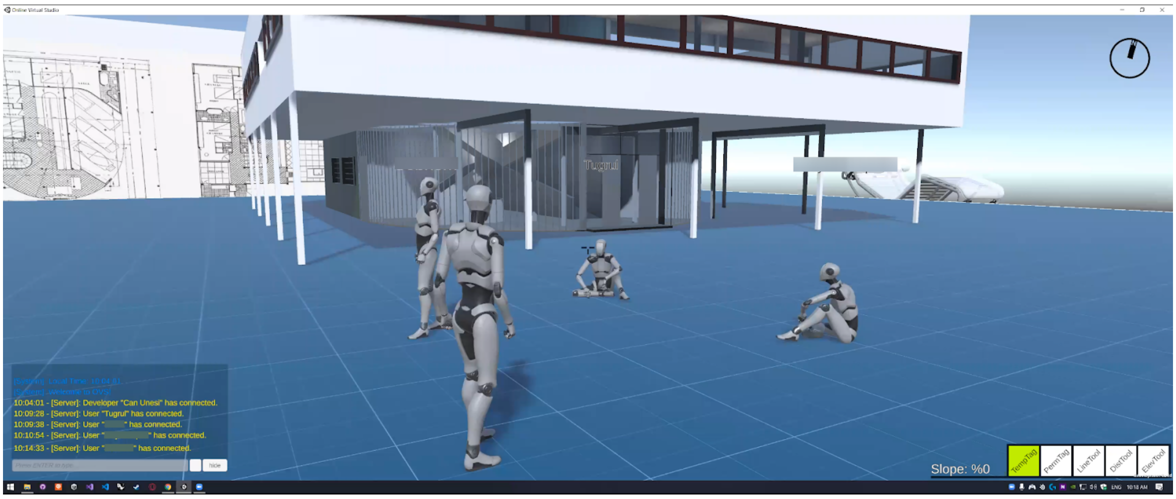

Figure 5.13. A Multi-User Session

- **Communication**

For communications, a text chat box was implemented, which also displayed system messages such as a welcome message, user connection and disconnection notifications. Voice and video communications were deprioritized and it was decided that a simultaneous video conference session could be held between the users until these capabilities were introduced.

- **User Interface**

A wireframe of the user interface was developed. Menus such as, main menu, connection menu, in-session context menu and in-session pause menu were developed. A basic toolbar displaying the active tool was placed in HUD.

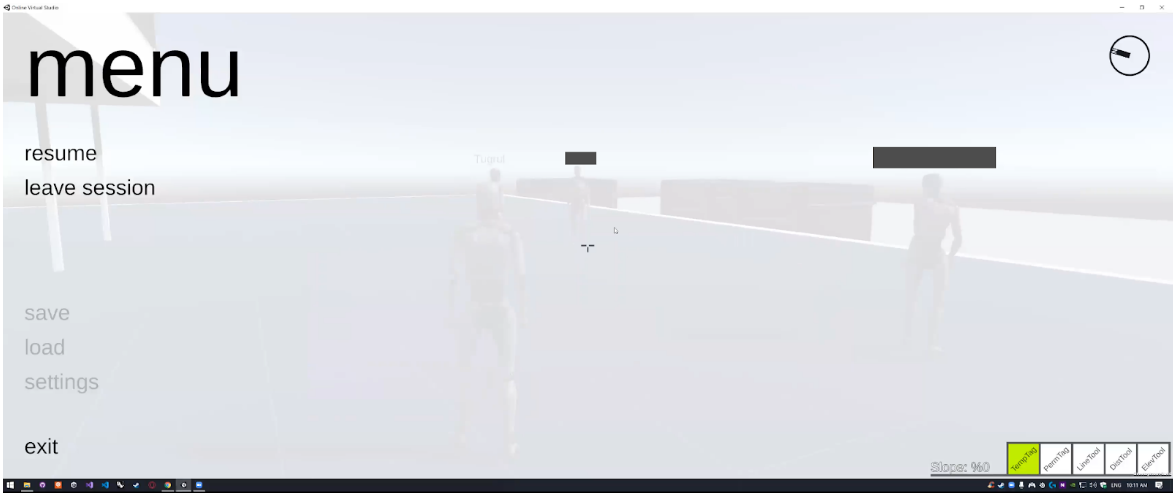

Figure 5.14. In-session Pause Menu

- **Annotation Objects**

Permanent markers were attached with the ability to type and store notes in them. The notes stored inside could be displayed as pop-up text on top of the markers.

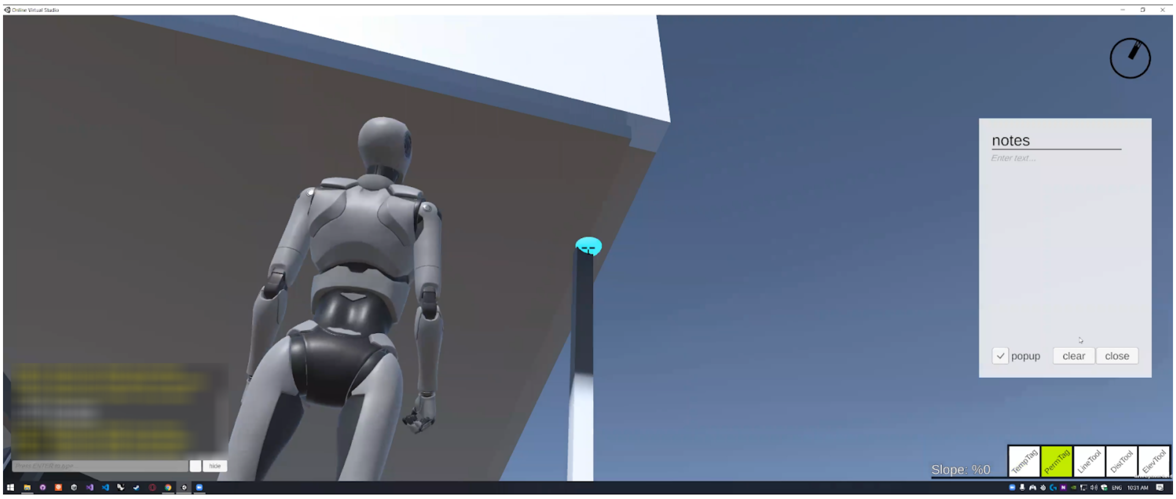

Figure 5.15. Taking Notes Inside a Permanent Marker

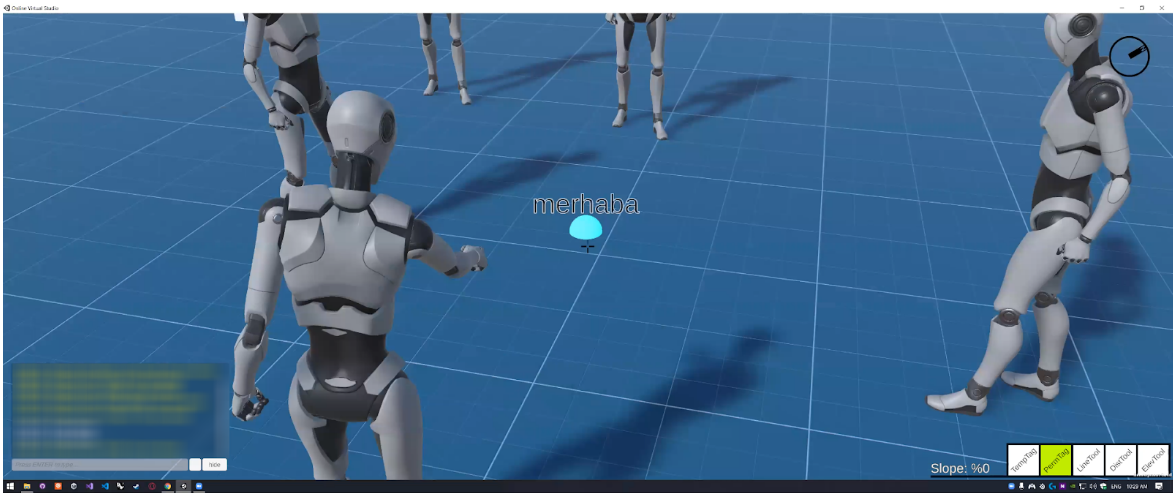

Figure 5.16. Pop-Up Note On Top of a Permanent Marker

- **Avatar Scaling**

A fixed implementation for avatar scaling was developed. This method utilized colliders to automatically scale avatars when they stepped on to certain objects. The scaling factor was predetermined on the object properties, e.g. stepping on a smaller model on a display stand made the avatar automatically smaller, jumping back out of the stand made the avatar bigger.

- **2D Assets**

Through utilizing display boards, 2D assets such as posters and drawings were able to be imported into the environment.

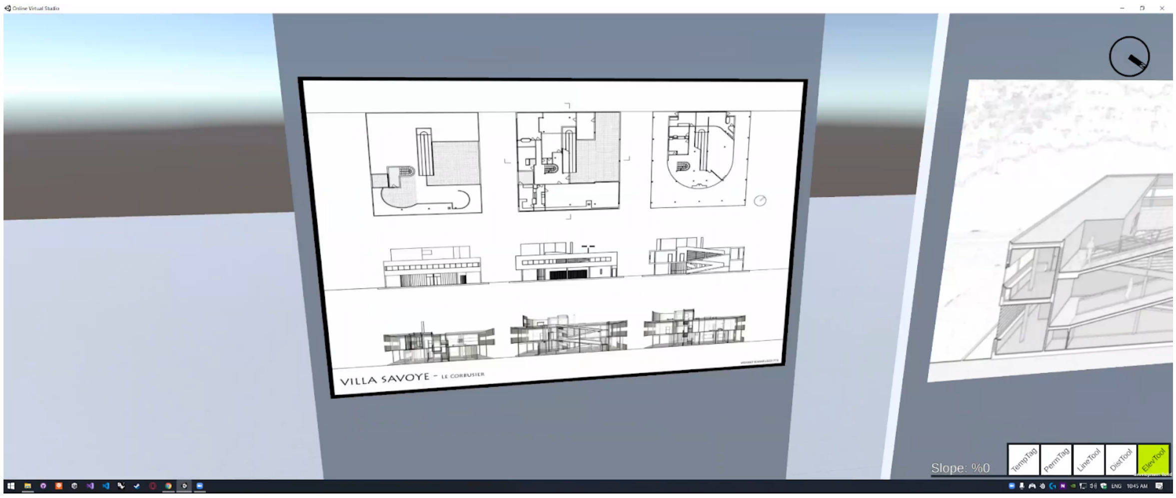

Figure 5.17. A Display Board With a 2D Asset

A desktop build of OVS was distributed to the tutors and they were asked to join the hosted session at the start of the meeting. Tutors were instructed on how to join a session, and the controls and features regarding the environment were explained once they joined.

One of the earliest comments was if avatar customization is going to be possible in the future. This comment later proved to be a requirement when participants moved away from each other and couldn’t read each other's name tags, stating that distinguishable characteristics of avatars could help in such cases.

Joined participants were welcomed with hand waving animations as a natural occurrence, which converted the meeting to a more game-like, entertaining environment. While the instructions were given, tutors even used sitting animations while listening, which might have been an indication of immersion.

The 3D environment was a built-in, static scene featuring Le Corbusier’s Villa Savoye. A 3D model of the building was placed on a display stand in the middle of the scene, with vertical display boards surrounding it. The display boards had various 2D drawings of the building on them. Participants started in micro-scale, adjusted to the scale of the 3D model on the display stand. When users navigated their avatars to jump off the display stand, the avatars were scaled up to match the scale of the display boards and the surrounding environmental objects such as the furniture Corbusier designed for the building, which were placed around the display stand.

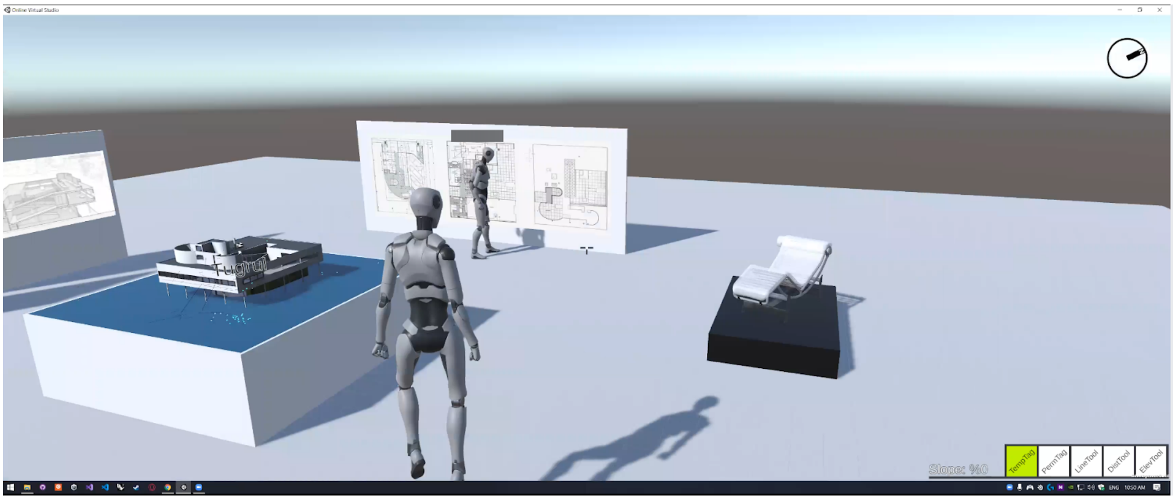

Figure 5.18. A 3D Model of the LC-4 Lounge Chair in the Environment

The aim of this meeting was to test the multi-user capabilities and question the overall usability of the software from the tutors perspective. The navigational features as well as annotation and communication tools were discussed, other features such as simulation and analysis tools were not examined.

The WASD scheme was found difficult by some tutors. They suggested that a more common control scheme, such as the pan and orbit controls, or a spectator view should be developed to avoid such difficulties with users who are less experienced in video games, in the future.

Regarding avatar usage in combination with colliders attached to both the avatars and the 3D models, the collisions were found positive in regards to navigating in a space while having the opportunity to question if multiple people would get stuck in certain spaces when moving adjacently.

While traveling inside the model, certain problems were found regarding visibility. A discussion about the probability of such issues happening in a design studio class was initiated as a result. The model used was a community made 3D mesh model of the Villa Savoye, which had some problems regarding mesh face orientations. It was explained by the author that in order to have correct visibility and proper collision, students will have to carefully construct or convert their models regarding face orientations. It was discussed that perhaps, a “proper modeling for virtual environments” lecture must be given to the students at the beginning of the semester, if not earlier.

After the introduction to OVS and how to use it, the environment was explained further. It was stated that this is a model done for a presentation scenario. After this statement, the participants were invited to jump off the model stand to have them scale themselves up automatically, and see the presentation setup.

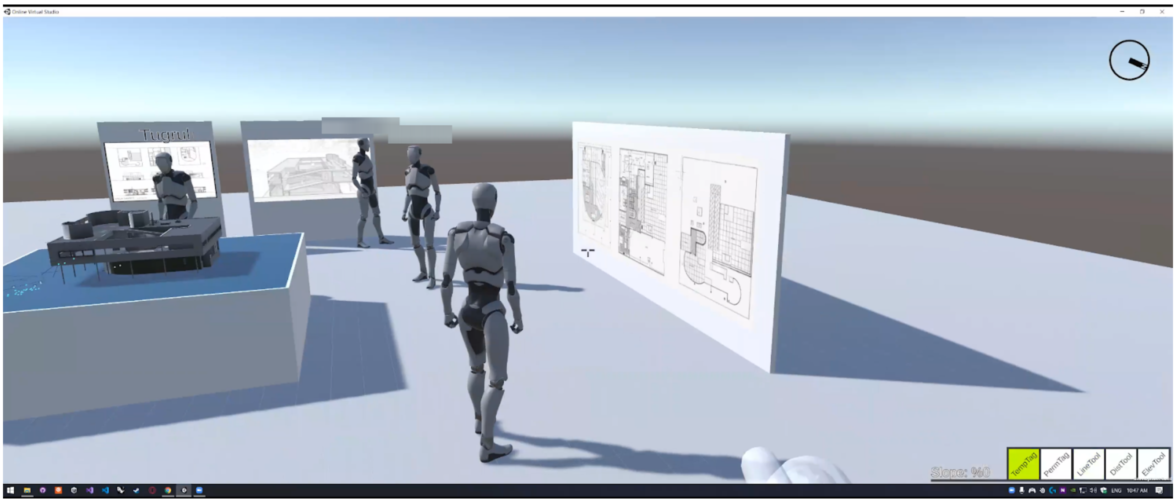

Figure 5.19. The Presentation Setup

After seeing the presentation setup, the tutors criticized this approach by asking if such a setup is still relevant in a digital environment. They added by stating that perhaps being in a digital environment, a conventional presentation setup was no longer necessary. They found that having a familiar setup might have a positive impact on the users in terms of getting acquainted with this foreign environment. However, they raised the question of whether it’s necessary to have a conventional “model on a table and posters on walls” approach in an environment where one could reproduce these drawings from the 3D model itself.

On the topic of presence and a social studio space, the tutors made jokes about meeting in this space to chat from time to time. Even though these statements were in an informal, comical sense, they were noted as an achievement towards the goal of producing a social space.

In regards to scale perception, controlling a visible humanoid avatar and having scale transitions were found positive. It was reminded by the tutors that this has been an issue among students, predating the distance education period and that using OVS could help them in improving the issues regarding scale. While the prepared build only supported two scales, 1/1 and 1/20, it was suggested that having multiple scale factors, or an adjustable one, could be also useful.

Tied to the discussions on scale and being able to jump into another scale and exploring a model, the topic of immersion was brought up. An object library containing environmental objects, and NPCs controlled by AI was suggested. It was discussed that on the contrary to a physical design studio where the scaled models are less life-like, in this digital studio the scaled models could be more life-like. This was explained through depicting a model where the trees would have dynamic leaves and perhaps some NPCs such as birds and cats moving around inside the model.

On the other hand, the need for diagrammatic information layers was also brought up. Although it was in development at the time and not implemented yet, Airsketch was demonstrated on a WIP build. Airsketch tool makes it possible to draw on a certain saved perspective, with the option to return to the said perspective and see the recorded sketch.

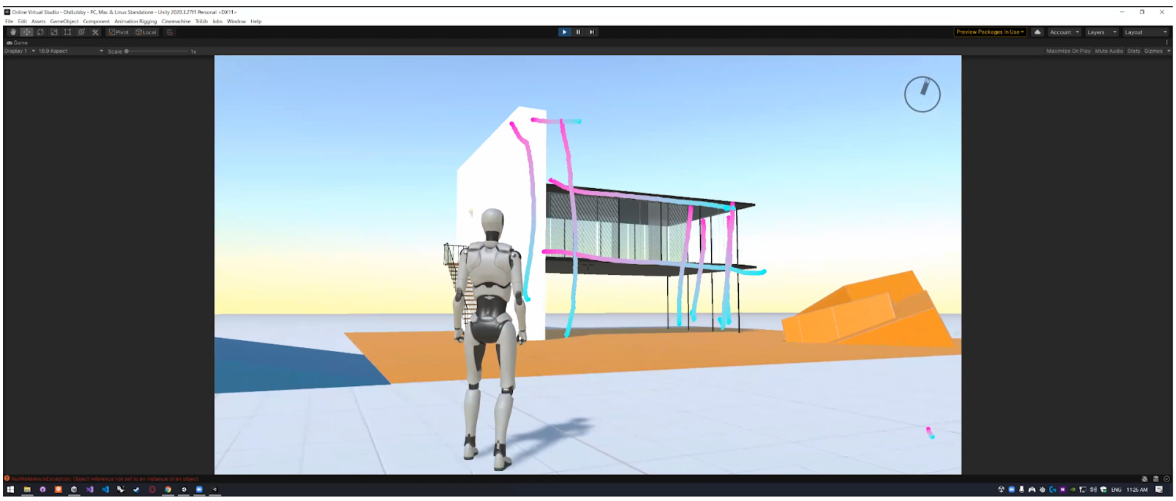

Figure 5.20. WIP Airsketch Tool

This feature was found necessary by the tutors and was prioritized in development after this discussion. At the time, it wasn’t carried into the multi-user branch yet.

5.3. Designing the Presentation Environment

After the last design studio interview, a considerable opportunity was found in using such software. The initial concept was to have the students import their models in a certain 3D studio template, with model stands and display boards. However, following the discussions and the unintentional developments, it was decided that the students should also design their own virtual presentation environments without restriction. After this point in development, any further development regarding scenes was ceased. The decisions regarding the methods and the environment used for their presentations, whether it's a more conventional approach such as model stands and boards, or a more creative one such as utilizing a possible exhibition hall in their projects or designing an unconventional one, was left for the students. It was only explained that this should be done through modeling the presentation environment that includes all the presentation material, with consideration for scaling opportunities, and importing the presentation model into OVS.

5.4. Test Environment

The ARCH 402 course of the Istanbul Bilgi University's Faculty of Architecture, namely, "Architectural Design VI", was chosen as the environment for OVS to be tested. Prof. Dr. Tuğrul Yazar, the supervisor of this research, and Nilüfer Kozikoğlu's joint studio group was selected to accept students for participation. The students of this group were asked to participate in the research on a voluntary basis, and they were asked to use OVS throughout the semester, if they chose to participate. The studio group was composed of 22 senior students and 2 tutors.

The studio focused on the coastline of Kadıköy, where there is a wastewater treatment plant on an infill site. The students were asked to design a community center for the Building Biology and Ecology Institute of Istanbul. Being a large site, an extensive architectural program was given to the students. The total area for the project was estimated to be 6500 square meters. The students were also influenced to discuss the urban transformation of the area, in regards to their project.

Concerning the pipeline for OVS distribution, the students were to be given a desktop client of OVS. As there hadn't been an updater developed yet, the students were expected to download the latest build when they were informed through a communication group.

As the importing features were developed before the start of the semester, the students would have the opportunity to import their material in runtime. The data format was chosen to be either OBJ or FBX, regardless of the type of the material. For the 2D material, they would have to place the 2D assets in a 3D scene. Due to the complications which improper models could bring, the students were given a lecture on OVS, and proper modeling and exporting for a game engine based virtual environment, which was also explained to be the same set of rules for modeling for VR, AR and the *metaverse*.

After the introduction, the students were asked to keep using OVS with their projects. The instances where the students used OVS in studio sessions were recorded with their consent.

5.5. OVS

Certain developments and improvements were made regarding OVS between the third design studio tutor interview and the term in which OVS was used. Below is the complete list of features and implementations that existed in OVS during this term.

- **Standalone Build**

As planned, OVS was built and distributed as a standalone application detached from the game engine, the Unity Editor application, and ready to be published if chosen to do so.

This approach is necessary to achieve multi-user capabilities and is the convention when distributing any form of application to the end-users. On Windows, OVS runs on a .EXE executable file format, which launches the application in a new window. OVS was also built for Linux and Mac, Linux being the platform for the dedicated server. Similarly, they run through executable file formats on these operating systems, namely ".x86" or ".x64", and ".APP".

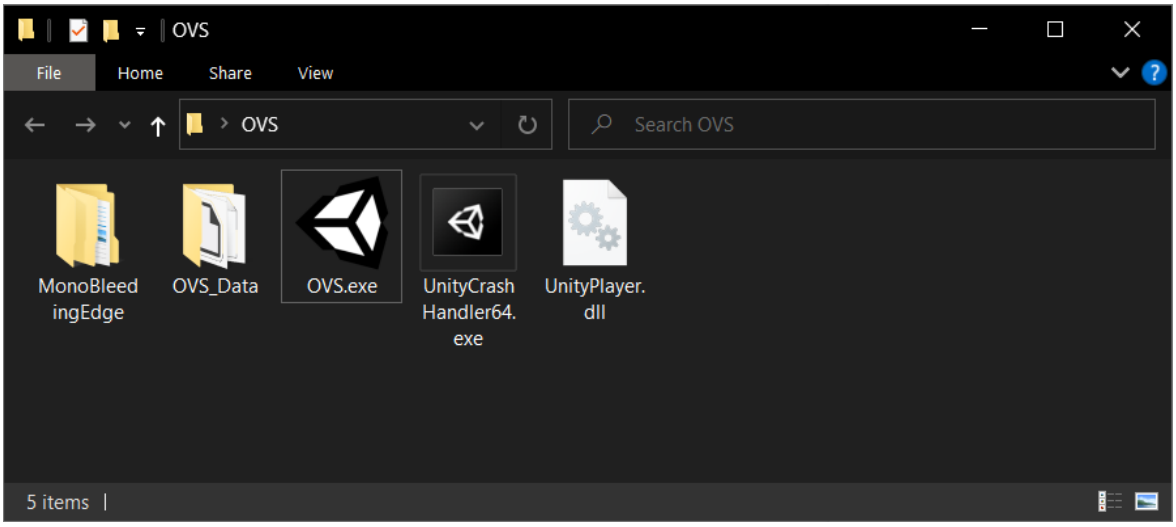

Figure 5.21. OVS Distribution

When "OVS.EXE" is executed, the application is launched and the users are greeted with a landing menu. The window of the application can be resized or made fullscreen. All of the UI programming was done especially to accommodate the free-sizing of the application window, making it possible to run it along with other windows in a multi-window working scenario.

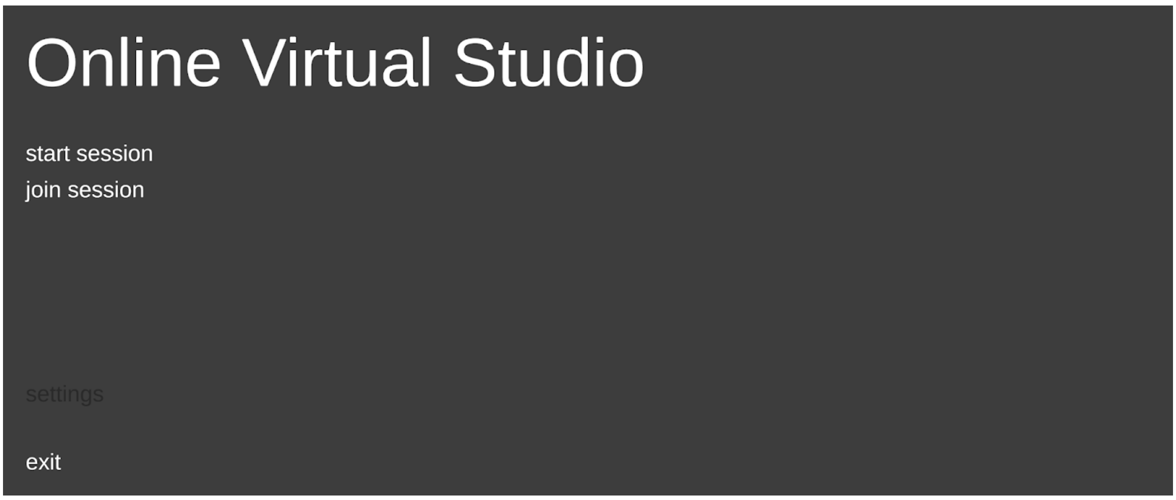

Figure 5.22. Landing Menu

- **Multi-User Capabilities**

OVS has multi-user capabilities, which work on either client-server architecture or through the dedicated server at Istanbul Bilgi University. Although OVS is aimed for multi-user scenarios, users can host their private sessions in which they can be alone until they invite other users. The “Start Session” button starts such sessions with the initiating user assuming the role of the host that the other users will be connecting to. Host handles all the connections and the data communication between the users, unlike the planned P2P approach, client-server architecture is a centralized one. There are plans to convert this architecture into P2P in the near future.

When a user starts a session, they are immediately taken into a blank environment with only a floor with a central blue patch that defines the origin, which also prevents the users from falling until they upload a model.

Figure 5.23. Blank Environment

Any other users who wish to join a session must know the IP address of the host of that session. An indexed lobby system with a session/server explorer window and short lobby codes are planned to be developed. Users who wish to join can do so by clicking the “Join Session” button, which takes them to a connection menu where they have to input the IP address of the host or the server, their name, and the password of the session if it’s a password protected one, such as the dedicated server sessions hosted from the Istanbul Bilgi University. Codenames for the dedicated sessions were defined and hardcoded to avoid typing IP addresses, although it should be noted that this approach requires static IP addresses.

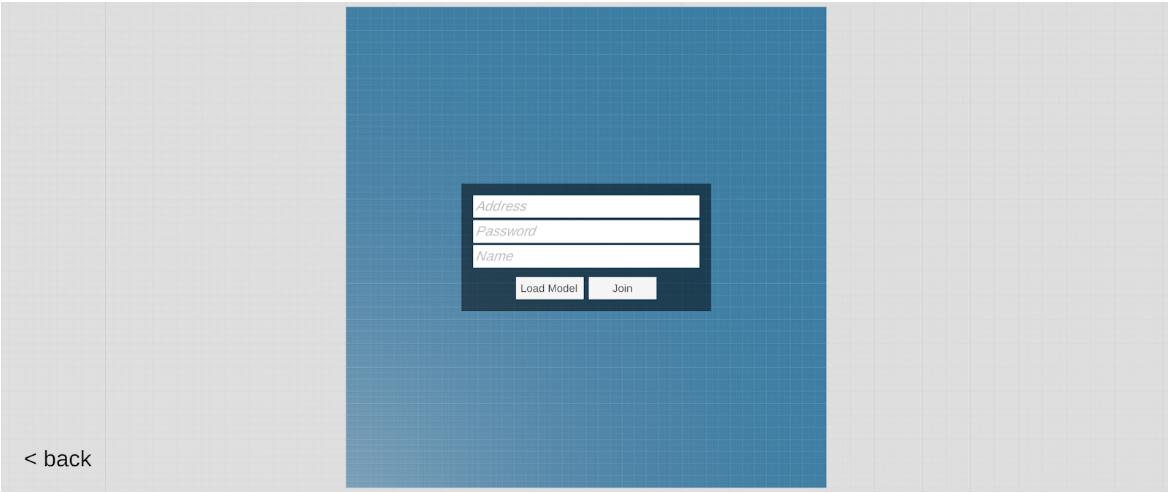

Figure 5.24. Connection Menu

A “Load Model” button is present in the connection menu to let users import the model that is being used in the session prior to joining to reduce the chance of disconnection timeouts while a model is being loaded if it is done when in an active session. Such loadings can freeze the running program temporarily, which may trigger an automatic connection timeout action by the host. When imported, the model is also visible from a top-down perspective view in the connection menu.

- **Model Importing**

Either prior to connecting to a session or after hosting or connecting to one, users can upload models in runtime. The file format can either be in .OBJ or .FBX, with or without materials or textures attached. If it is to be done while inside a session, the in-session context menu is employed to access the importing functions.

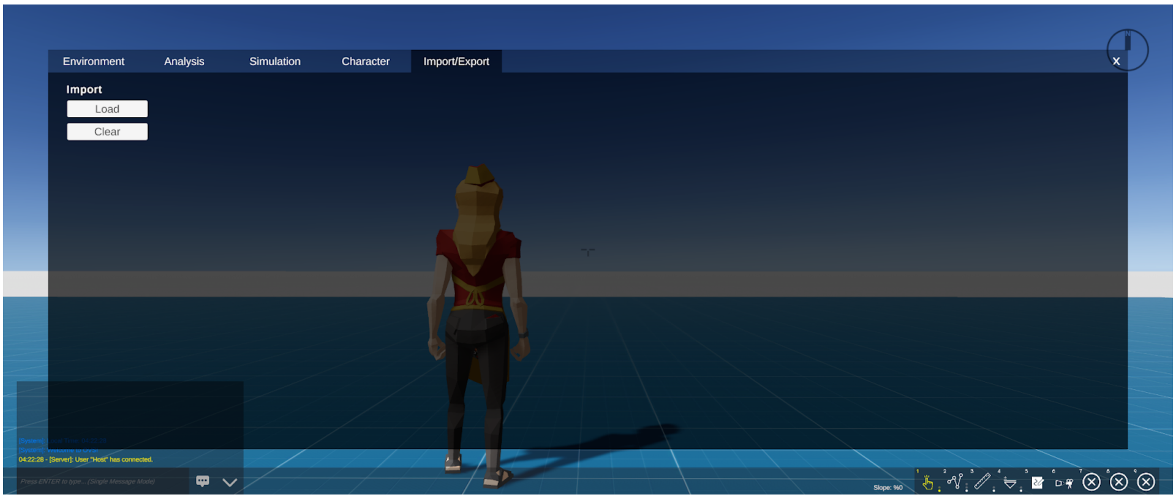

Figure 5.25. Import/Export Tab in In-Session Context Menu

To reduce network loads and certain issues, model importing was kept asynchronous. Which means that each user is responsible for themselves in importing a model. This also means that users can import separate models into the environment, with no way of determining what users have imported. In the near future, a FTP server will be employed to keep track of the models and share them between users, disabling the asynchronous approach.

After a model is imported into the environment, it is placed according to the origin it was exported from, to the origin of the environment, which is the center of the blue center grid in the initial blank environment.

Figure 5.26. Loading Model in Progress

The percentage of the loading can be tracked and after the model is loaded completely, the floor of the initial environment disappears.

The models are scaled through units that they were drawn in. In this period, the participants were asked to export their models using meters as the model unit, due to the fact that OVS uses meters as units. As an example, if a model was made in centimeter units and exported into OVS without adjusting, the model would be imported in 100/1 scale. All imported models are attached with colliders, which are calculated while being imported.

Figure 5.27. Imported Model and Environment

- **HUD**

A heads-up display is employed to keep track of the compass, chat box and the toolbar. A 3-line reticle is also present in the center of the screen, which transforms into a 4-line one when the user zooms in on the view. Holding the F1 key displays a help overlay, showing the controls on a visualized keyboard and mouse. The F2 key disables the HUD for clean visuals, allowing for taking screenshots without the HUD.

Figure 5.28. Help Overlay

- **Menus**

Pressing the ESC key displays an in-session menu with the options to resume, leave session, return to main menu (landing menu) and to exit the application. Accessing application settings such as graphics and controls options will be done through this menu when implemented. Saving and loading sessions will also be accessible here.

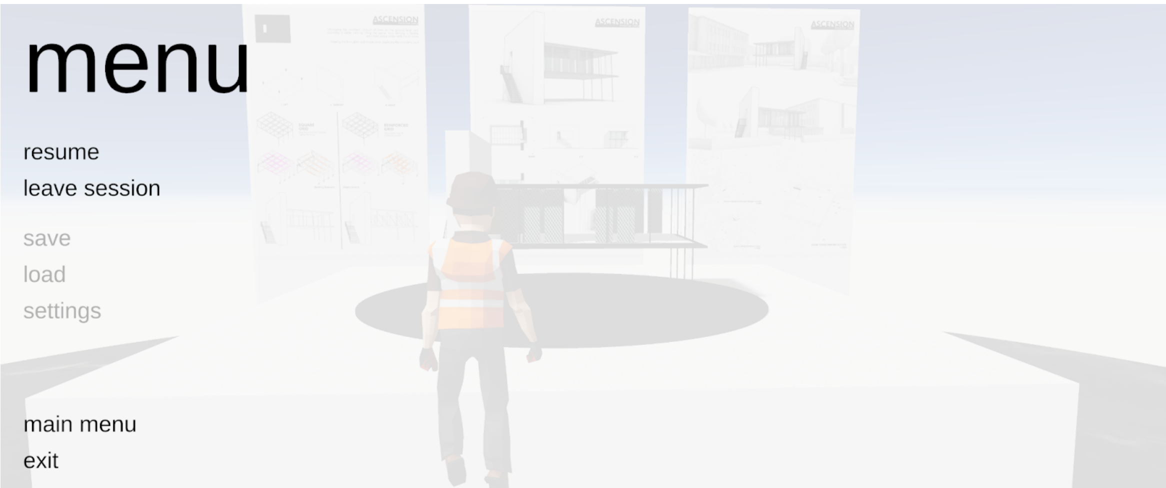

Figure 5.29. In-Session Menu

Exiting the application launches the feedback survey, which the users are asked to participate in after each use.

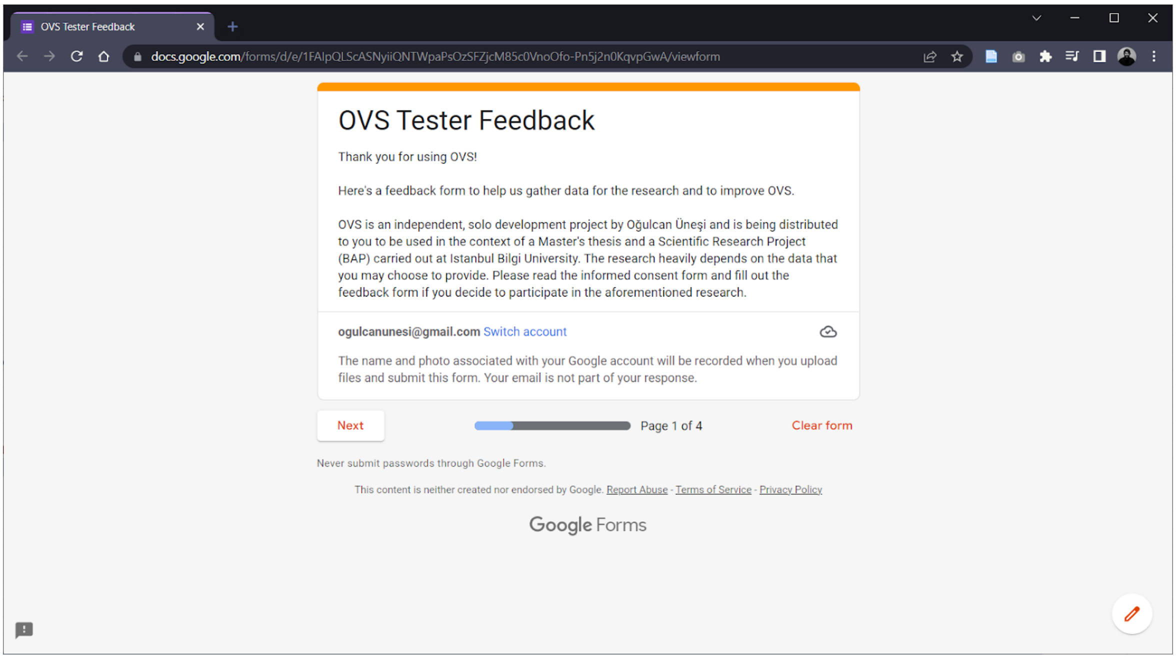

Figure 5.30. Feedback Survey

An in-session context menu is accessible through the M key, which holds features such as simulation and analysis methods, as well as the aforementioned import/export capabilities.

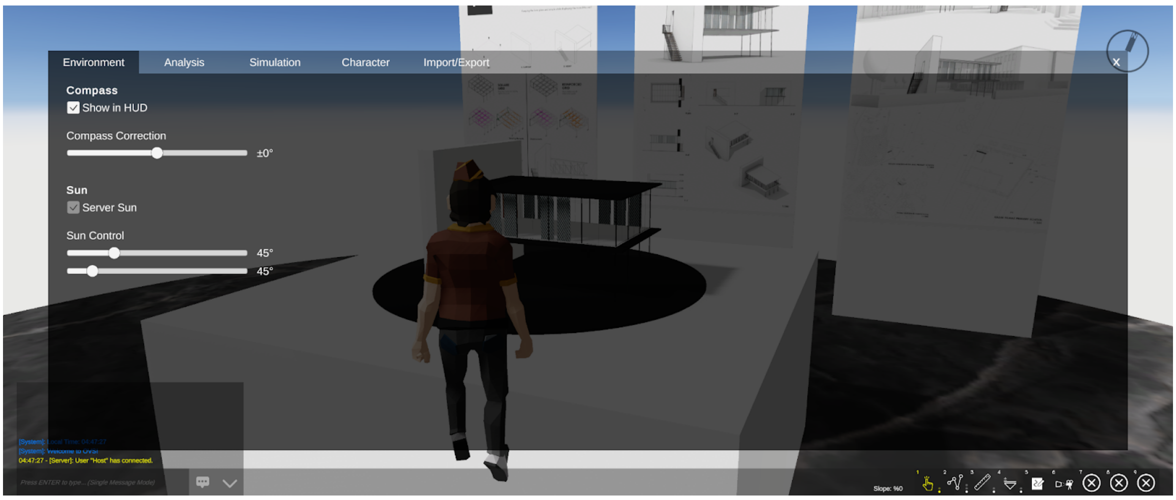

Figure 5.31. Context Menu

The context menu consists of 5 tabs, namely "Environment", "Analysis", "Simulation", "Character" and "Import/Export".

- **Environment, Analysis and Simulation**

Currently, the “Environment” tab holds options regarding the compass and the sun in the environment. The compass can be displayed or hidden through this menu, or it can be corrected with a slider that introduces additional rotation in case the model wasn’t exported with north orientation in mind.

When a user is the host, they can control the sun orientation of the server, which is also transferred to the connected users. The users who wish to control the sun on their own can disable this feature by unticking the “Server Sun” checkbox, which can be re-enabled and synced when ticked.

Analysis tab consists of options for slope display, user trails and gaze heatmap, with gaze heatmap being disabled for being unoptimized in terms of performance. User trails can be recorded, shown and cleared. This feature is kept on client-side to enable for individual tracking periods.

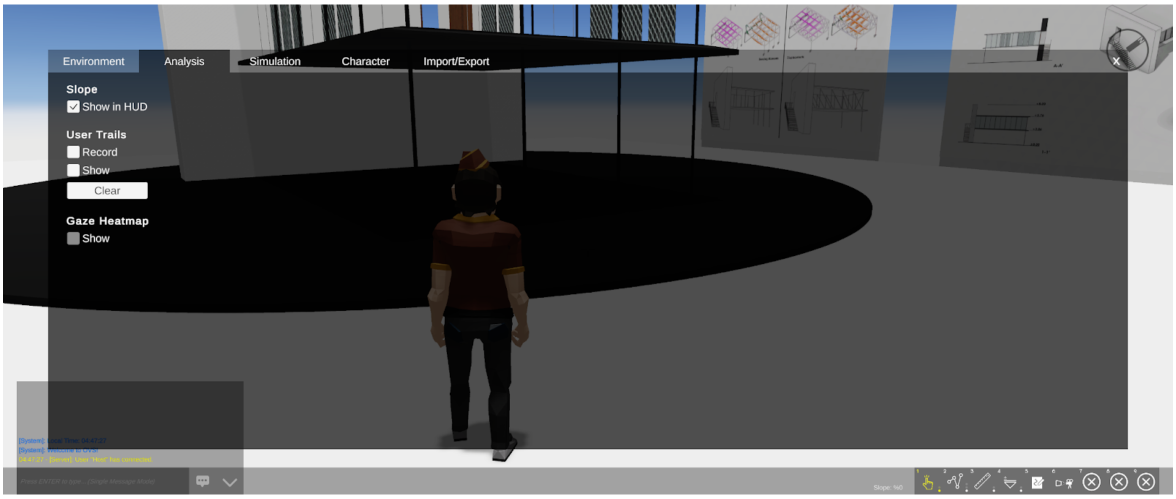

Figure 5.32. Analysis Tab

When the “Record” checkbox is ticked under “User Trails”, all of the connected user positions are being recorded at distance intervals, which can be displayed as polylines with a color gradient to indicate the start and the end of the travel.

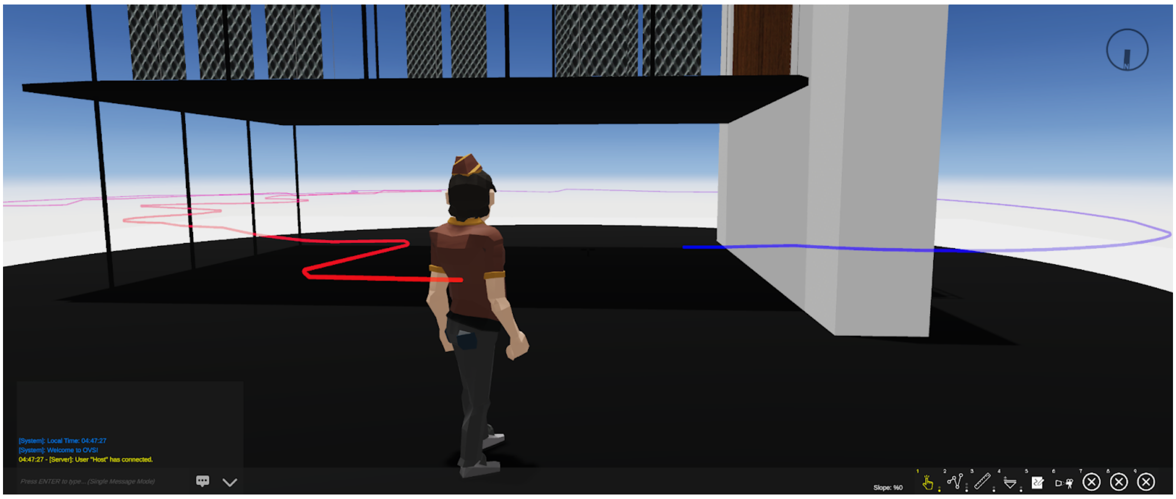

Figure 5.33. User Trail

The "Simulation" tab consists of the basic rainfall simulation through spawning physics-based prefabs, called "Rainballs", and the wheelchair simulation options.

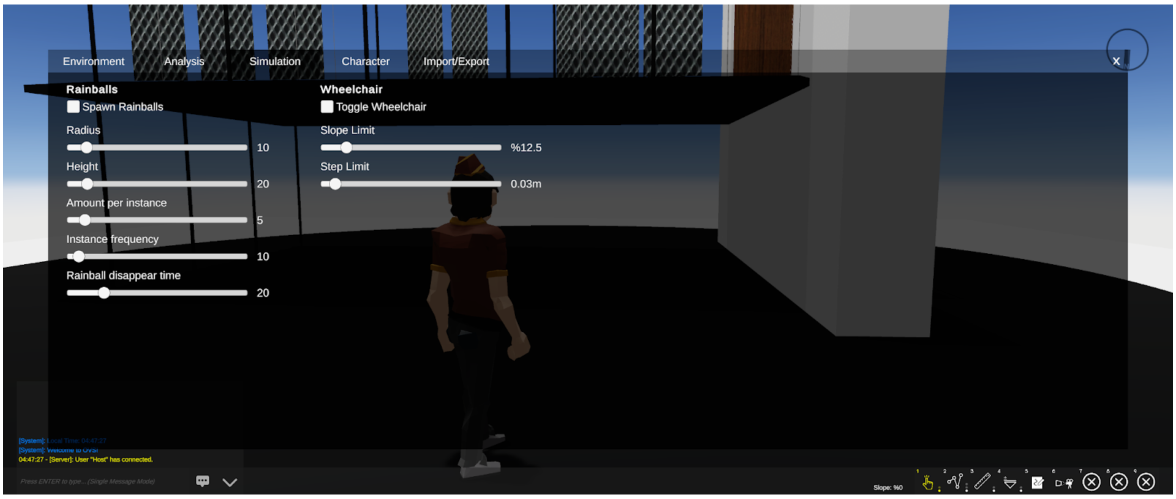

Figure 5.34. Simulation Tab

Rainballs can be spawned within the set radius originating from the spawning user, with options for spawning height, spawned amount per instance, instance frequency and disappear (despawn) time. Wheelchair simulation options consist of toggling the wheelchair, and adjusting slope and step limits.

Figure 5.35. An Avatar on a Wheelchair with no HUD

- **Avatars**

As previously explained and discussed, OVS employs the use of avatars in a 3D environment. The user camera is attached to an avatar, which is controlled through the WASD scheme. Users/Avatars can walk, run, jump and fly through this control scheme. While on ground, the avatars collide with each other and the environment, however while flying, they can move through objects. If a user falls through the ground and continues falling for a period of time (5 seconds), they are sent flying back to the level of origin to avoid incapacitation. Each user has their own avatar customizable through the context menu, and with a name tag visible to other users.

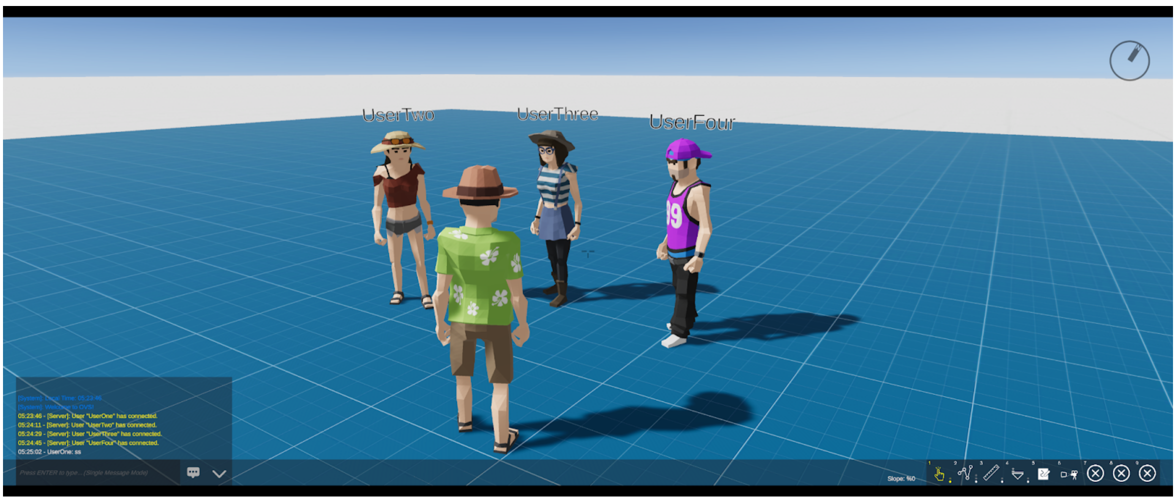

Figure 5.36. Various Avatars with Name Tags

Name tags always face the viewing users' camera independently (client-side), and get smaller with distance to avoid cluttering of the screen space. Various avatar models were employed to make the users distinguishable from a distance. All avatars are animated and the animations are synced.

The animations consist of the ones associated with movement, procedural ones such as head rigging to face camera direction and pointing, and social ones such as waving, dancing and sitting.

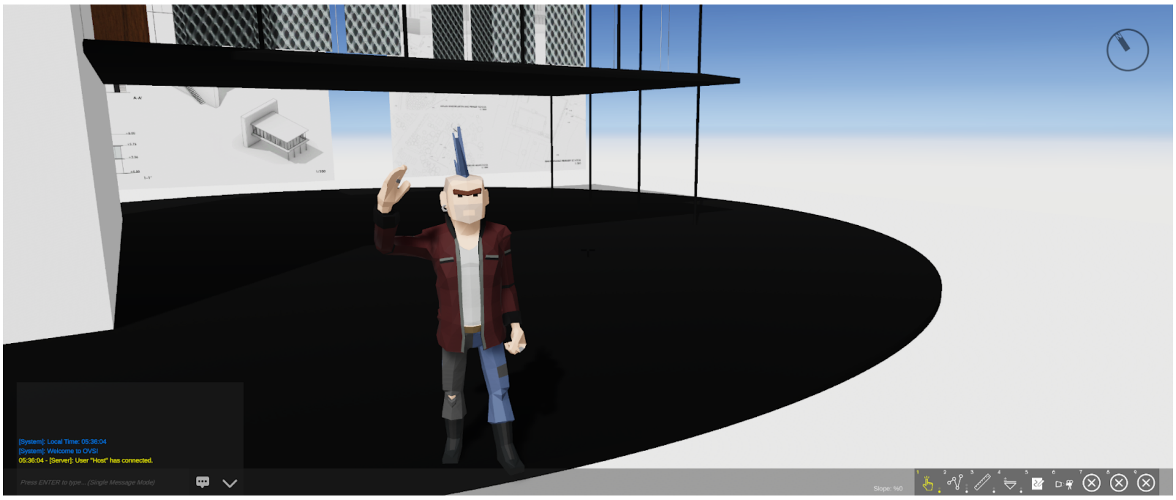

Figure 5.37. Waving Animation

The avatars are individually scalable, with a predefined scale, which was adjusted from time to time during the term in which OVS was used in the architectural design studio classes. A method of adjusting the scale factor will be implemented in the future.

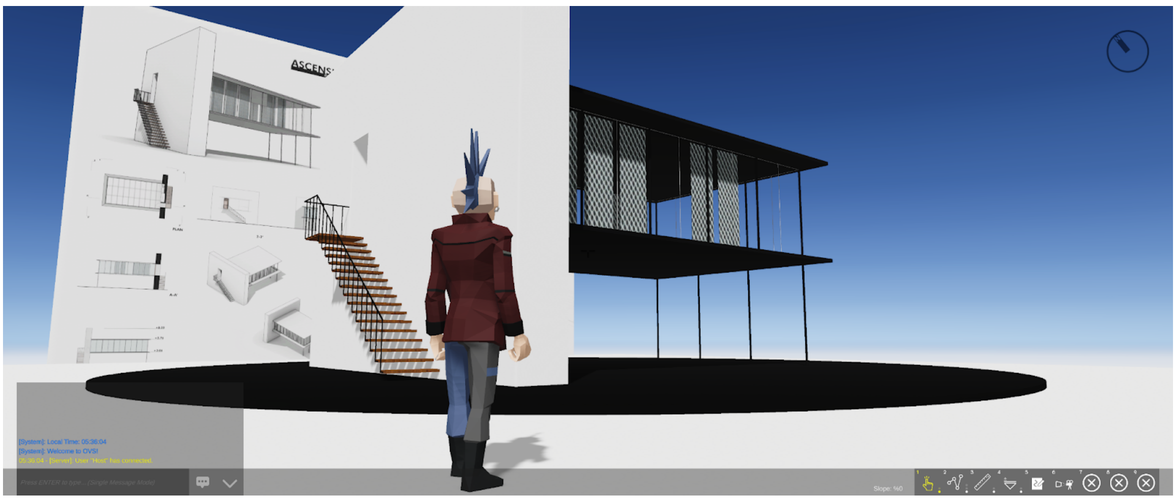

Figure 5.38. User Avatar in 1:1 Scale

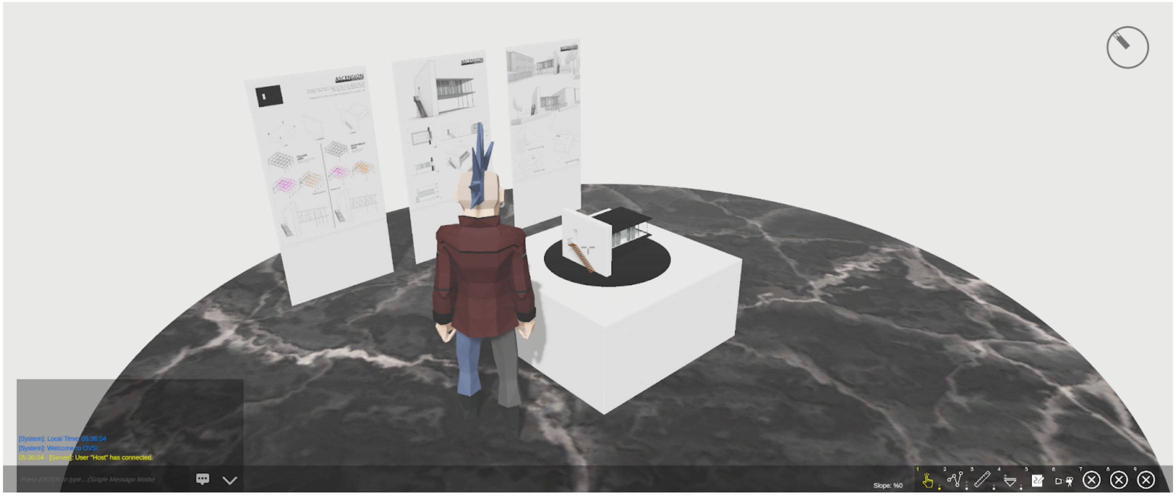

Figure 5.39. User Avatar in 20:1 Scale

The scaling is done on the avatars, rather than the imported models to ensure that all users can view the environment and the model in various scales, simultaneously.

- **Camera**

Initially, the users view from a third person perspective camera, behind their avatars. The camera is also offset horizontally from the avatar to let users see what's in front of their avatars. This offset can be toggled to be either to the left or the right side of the avatar to avoid obstructions that may result from the camera collisions.

The camera collides with the environment and always keeps the user's avatar in view while not flying. However, while flying, this collision is toggled off. As in while flying, walls and floors may get in between the camera and the avatar. This is done intentionally as they are also able to go through objects, and to let users explore tight spaces with avatars, where the camera collisions may make it harder to navigate in a third person perspective.

Optionally, users may also switch to a first person perspective, viewing from their avatar's eyes. All controls and features exist within this mode as well, the only difference being the camera position and the field of view. In the situation regarding tight spaces, first person perspective is found necessary to be able to have walkthroughs in such spaces.

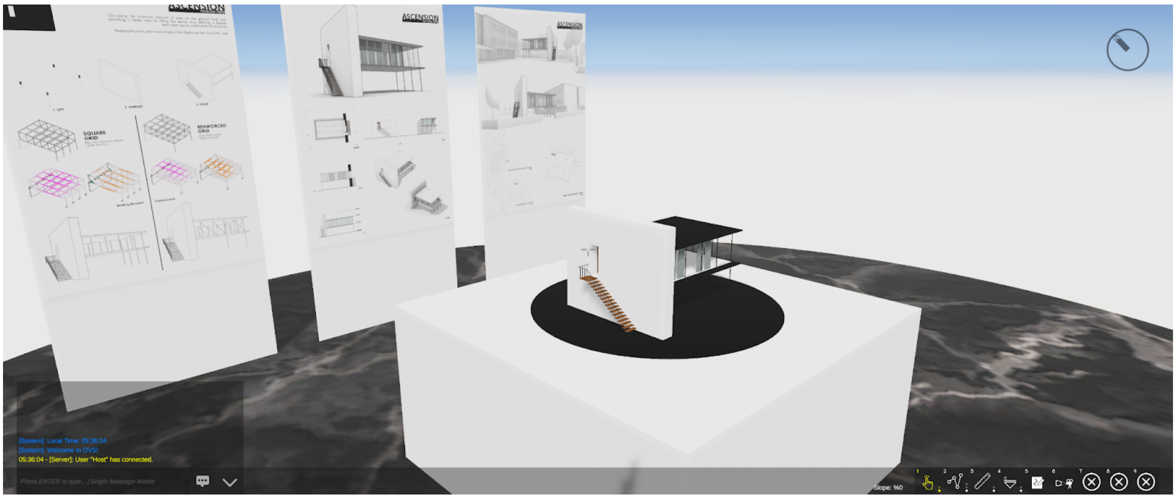

Figure 5.40. First Person Perspective

In addition to the perspective cameras, a parallel projection camera can be employed through the toolbar. In this view, the avatar controls are disabled and swapped with pan and zoom controls. The parallel projection camera can be spawned perpendicular to a surface, through the use of the corresponding tool in the toolbar. This method can be employed to have architectural elevations, plans and sections from the imported 3D models, or to view 2D assets such as posters. Section shaders were not implemented, but are planned to be developed.

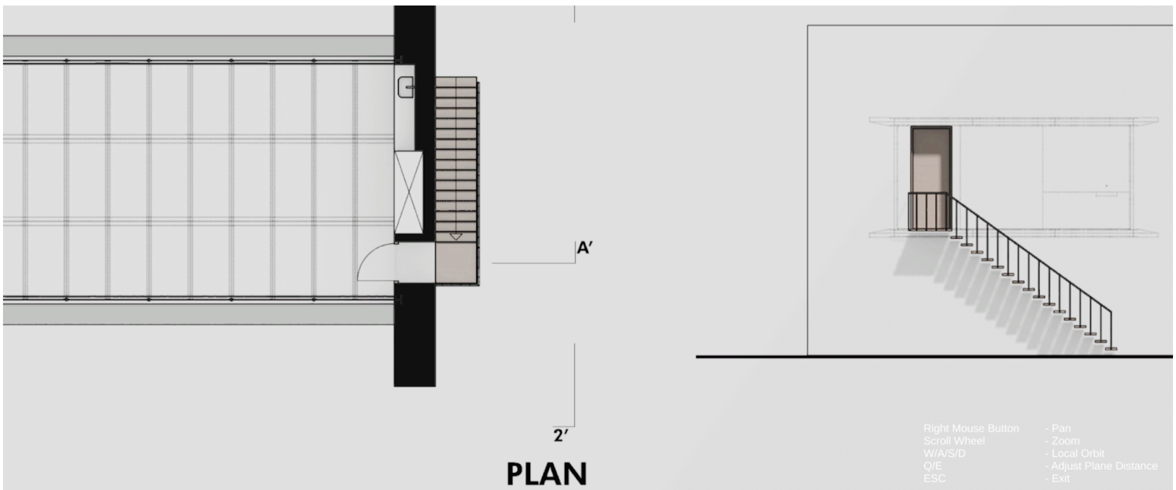

Figure 5.41. Parallel Projection Camera on the Posters

- **Communication and Annotations**

In addition to procedurally animated avatars and pointing, markers and annotation objects were developed for analysis and communication.

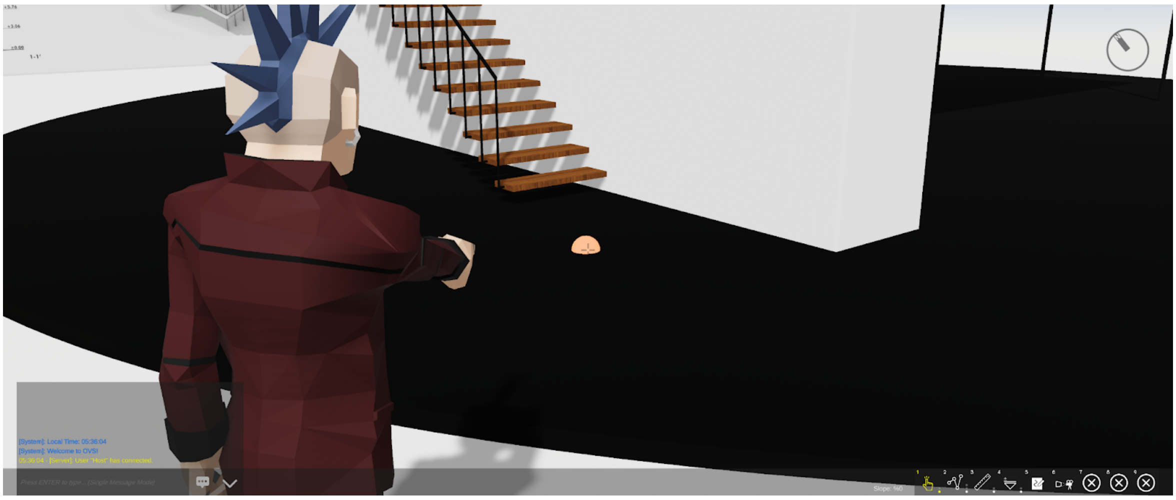

Figure 5.42. Pointing and Placing a Temporary Marker

The annotation objects are accessible through the toolbar, which is displayed in the lower-right corner of the screen in the HUD. Toolbar objects are selected through the keys 1 to 9.

All annotation objects can only be spawned while pointing, which provides workspace and consequential awarenesses. As an example, if a user wishes to place an elevation tag, they must first hit the “4” key, see that they have selected that tool in the toolbar through the highlighting, then point through holding the right mouse button, and place the marker through clicking the left mouse button while still holding the right mouse button.

Currently, there are 6 annotation objects and the parallel projection camera in the toolbar. Similar objects are also layered in toolbar hotkeys, such as temporary and permanent markers. This layering works through hitting the hotkey multiple times, for example, when a user hits the “1” key for the first time, they switch to the temporary marker, if they hit the key for a second time while still in the first slot, they switch to the permanent marker, which is also displayed through the change of toolbar icons and a slot layer indicator on the toolbar.

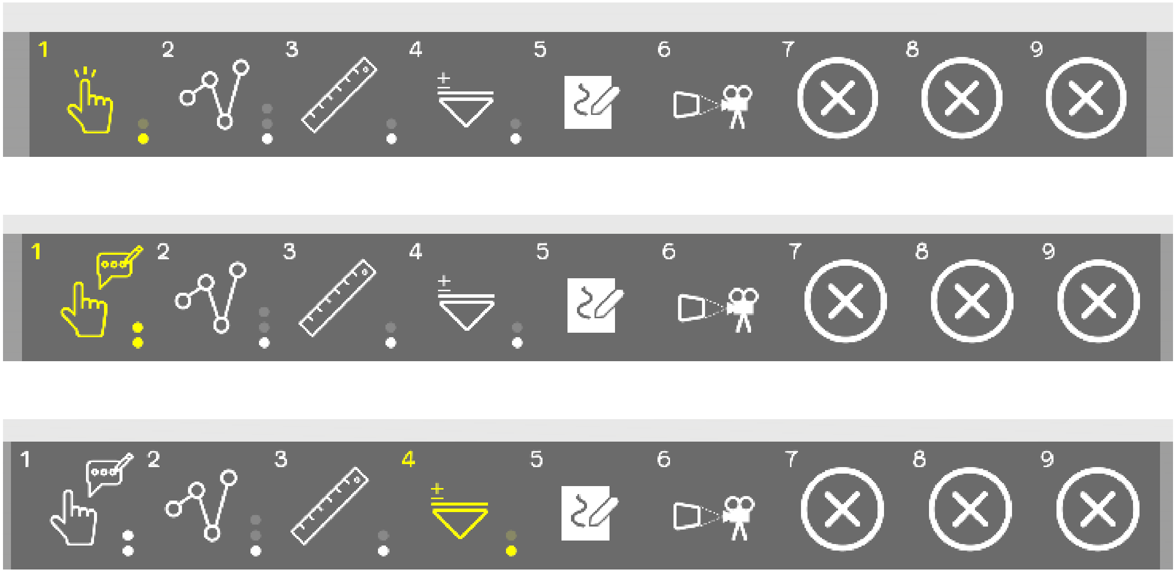

Figure 5.43. Toolbar (a) Temporary Marker Selected (b) Permanent Marker Selected (c) Elevation Marker Selected

Temporary markers are for pointing, they are spawned once per user unless despawned and change location on prompt if there is an existing one.

Permanent markers are for permanently marking a point. A user can spawn infinite permanent markers, with the option to delete them. Text can also be stored inside these markers, which can be displayed as pop-ups optionally.

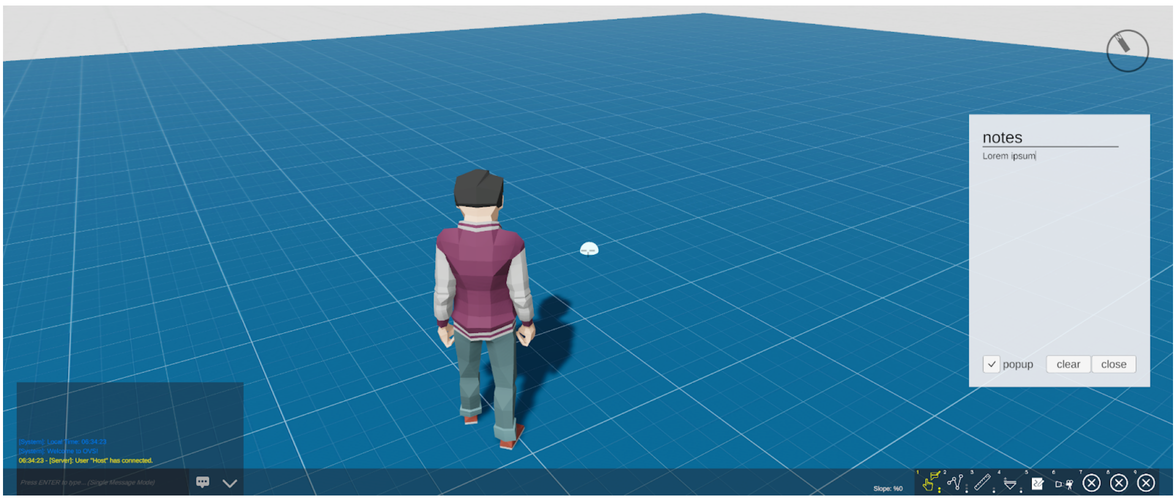

Figure 5.44. Typing a Note Inside a Permanent Marker

Figure 5.45. Pop-Up Text on a Permanent Marker

The permanent markers display a confirmation box while deleting, only if they contain text information. Users can delete any permanent markers, regardless of ownership.

Figure 5.46. Permanent Marker Delete Confirmation

Aside from annotation markers, 3D lines can be drawn and measurements can be made, all of which can infinitely be spawned and deleted by any user. 3D lines can be drawn between points, which can be snapped to create polylines. Distances can be measured through spawning consecutive distance measuring points, which despawn after the distance is calculated. Elevation tags however, depend on the initial spawned marker. The first marker is marked as "0" and unless all elevation markers are deleted, that point is considered in defining the elevation of the following points.

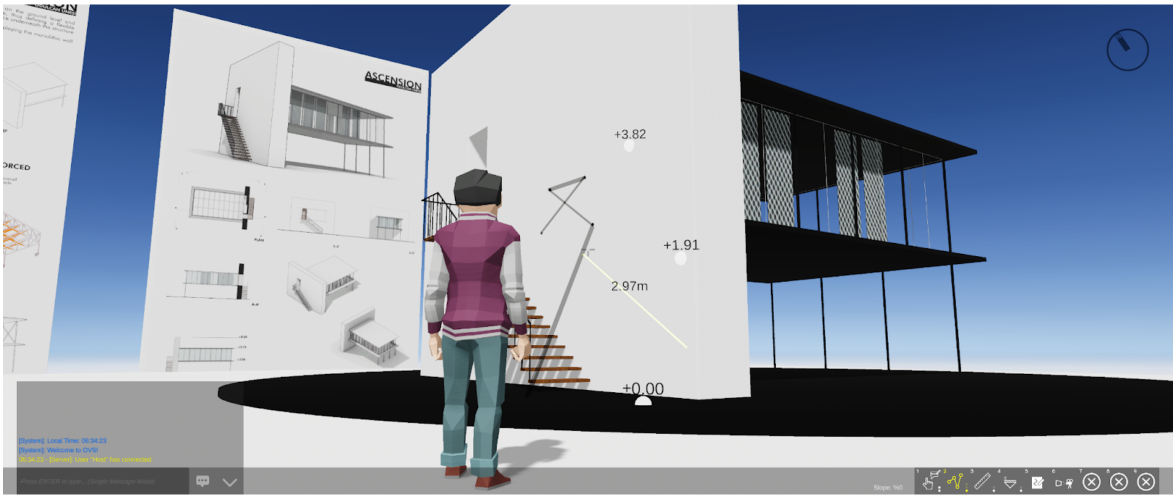

Figure 5.47. 3D Polylines, Distance Measurement and Elevation Markers

One of the most requested features, superposed sketching was developed in this period. Through pressing the “5” key, the user can initiate the “Airsketch” mode where they can draw on the screen. These drawings and the camera properties are stored inside a spawned Airsketch marker, which are hidden outside of the drawing mode. Through the markers, the sketches can be revisited or viewed by any user, even simultaneously. It is possible to hide the player model(s) while drawing to avoid visual obstruction.

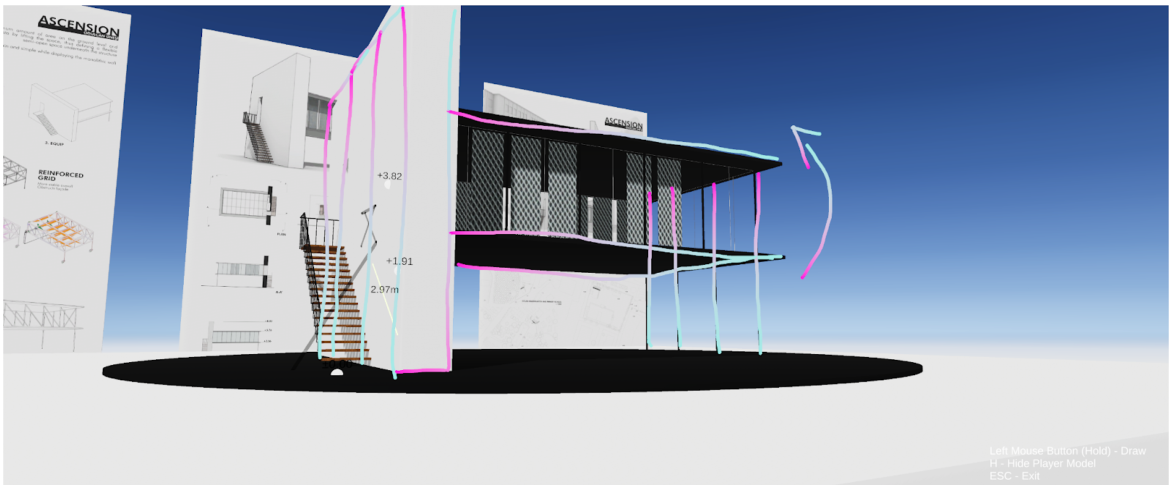

Figure 5.48. Airsketch Tool

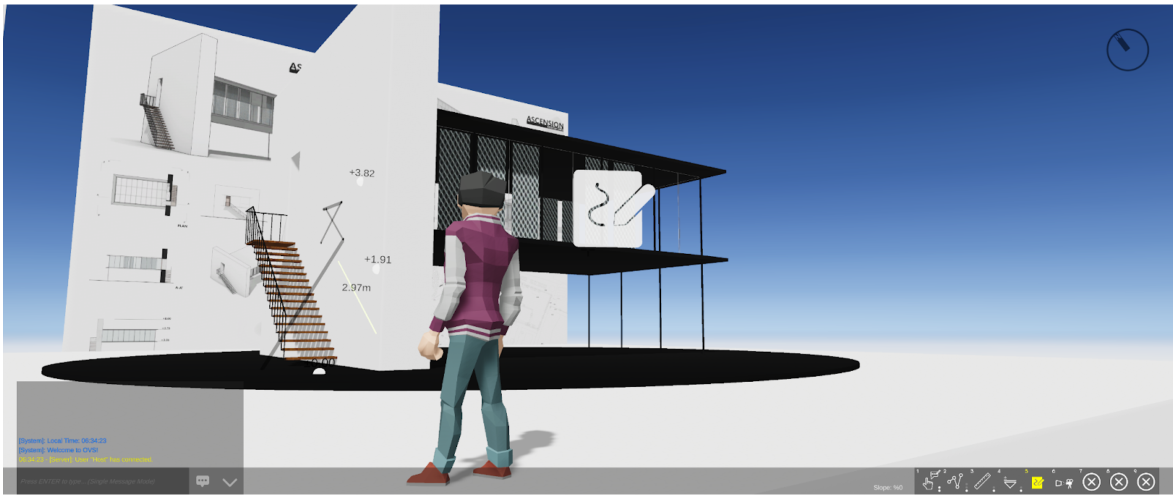

Figure 5.49. Airsketch Marker

5.6. Studio Sessions

Following the developments and the integration of the aforementioned features, OVS was out of the prototype phase and in the Alpha stage. In its current state, OVS was ready to be tested by users, the students and the tutors of "Architectural Design VI" studio led by Tuğrul Yazar and Nilüfer Kozikoğlu. The studio took place in the spring semester of 2021/2022 and lasted 14 weeks, twice a week. Starting from the first week, students were introduced with OVS and were asked to use the environment when possible. The usage of OVS was kept on a participatory basis and wasn't mandated. However, the students were given a lecture on the usage and there were times where OVS was preferred in certain situations.

Starting with the first session, the students were given a tutorial for OVS in the computer lab. in Istanbul Bilgi University. In addition to the explanation of the environment itself, the features and the controls, a lecture was given on proper modeling for virtual environments. In this lecture, the basics of mesh model behavior and operations, proper face orientations, nurbs to mesh conversion, and exporting in FBX and OBJ formats were taught. The students joined the session hosted on the tutor PC to experience the multi-user features, and then were left to use OVS on their own. Many connection problems occurred during this session and the development shifted its focus from feature implementation to bug fixing.

Figure 5.50. Introduction to OVS

Following the lecture and bug fixes on the netcode, a OVS was distributed to the students as Windows and Mac clients. In the second week of the studio, one of the tutors was infected with COVID. Although unfortunate, this posed an opportunity to test OVS in a distance education scenario. In this session, a large but simple model of the project site was imported into OVS and a virtual field trip was held. Although more students were able to connect this time, the connection problems persisted and the site was explored with separate groups of students instead of the whole studio group.

Figure 5.51. Virtual Field Trip

During the virtual field trip, the students were encouraged to experience the site in both scales (1:1 and 20:1) to be able to grasp the actual size of the site, as well as the physical qualities of it and the relation to the surrounding areas. Annotation markers, scaling and sun rotation were used during this session. It was observed that both the tutors and the students preferred to be in the larger scale when exploring the site, either due to the size of it or the lack of detail in the model which is more apparent when in 1:1 scale.

During exploring the site, one of the tutors expressed that they felt lost and anxious when they couldn't locate other participants. They added by suggesting that a way of locating the rest of the users should be developed, which was interpreted as a request for a map or a minimap that could show other users' location as well as the immediate vicinity. The development of a minimap was prioritized after this suggestion, however due to the focus on fixing bugs on the existing features and most importantly, to provide a more stable connection for the users, the development never reached the point for further feature implementations during this term.

In the following weeks, OVS was barely used by the students as their designs were still underdeveloped for producing 3D models. Although the environment could be used to present 2D assets as well, the need for a virtual environment wasn't felt at this stage as the education was face-to-face.

During the first midterm, the submission of 3D models of their projects was kept mandatory for the students. Despite not directly being related to OVS usage, this was seen as an opportunity to use these models in OVS and the students were told that they could present their projects in OVS during the jury. The midterm jury was held in two sessions, one being face-to-face and the other remotely.

Most of the models that were produced for this submission were not meeting the criteria that the students were taught during the first week. As a result, not all of the models were suitable to be used in a virtual environment. One group however, had made a model meeting the criteria, and chose to present their work in OVS during the face-to-face jury.

In this face-to-face setting, the students projected their screen and presented a walkthrough of their project in real-time, to the jury members. The ability to present a real-time walkthrough was found positive as the students had the opportunity to take the jury members to positions and perspectives that weren't displayed on the physical posters. However due to the model lacking much detail, interior walkthroughs were not found to be meaningful and as per the jury members' request, the project was observed from a flying perspective. While not utilizing any multi-user capabilities, this instance indicated a potential for a hybridized presentation environment, featuring virtual environments.

Figure 5.52. OVS in a Face-to-Face Setting

In the second session of the jury, a dedicated server build was utilized and a server provided by the Istanbul Bilgi University, for the scientific research project that was being conducted in parallel to this research, was used to host the session. A video conference call was held in parallel to OVS to provide communications, and the students were let into the server in groups to prevent connection problems. In this session, some of the student models featured their posters as 2D assets inside their buildings, in presentation settings. In addition to the previously used features, the parallel projection camera tool was utilized to inspect these posters.

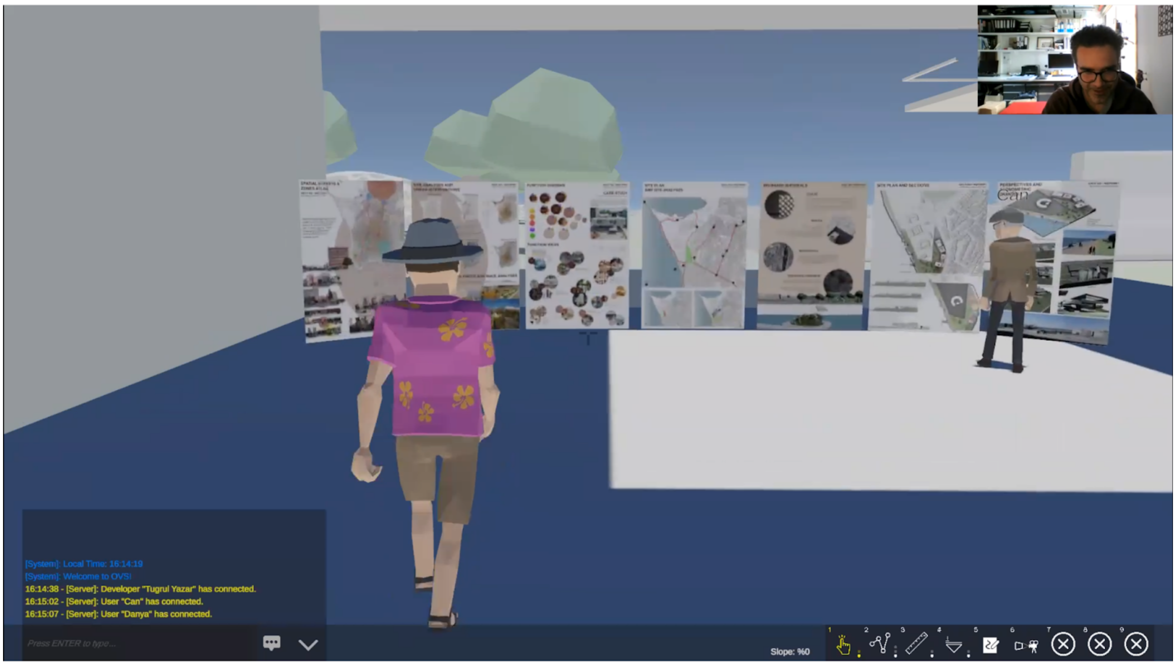

Figure 5.53. Posters Inside a Project

As most of the models lacked details, textures and materials, it was found hard to distinguish the edges in the environment. This brought the necessity of an edge detection shader which could help with such models through highlighting model edges, or a material overlay shader which could help through applying a more legible texture on the surfaces.

In the following weeks, OVS was again not used by the students during studio hours. As the time for the second midterm jury came, the students were asked again to submit 3D models of their projects, keeping in mind the possibility to present in OVS. The second midterms were held entirely remote. However this time, 6 projects were presented in OVS. The quality of the models were again not completely meeting the criteria, although they were better suited to be experienced in a virtual environment. On the contrary to the previous sessions, annotation tools were used more efficiently as the students were more acquainted with the environment and for the first time ever, the airsketch tool was used by the tutors to give criticism. Additionally, as the models were more detailed and included furniture at this stage of the designs, some out-of-scale objects and proportions were immediately recognised through the use of a human sized avatar.

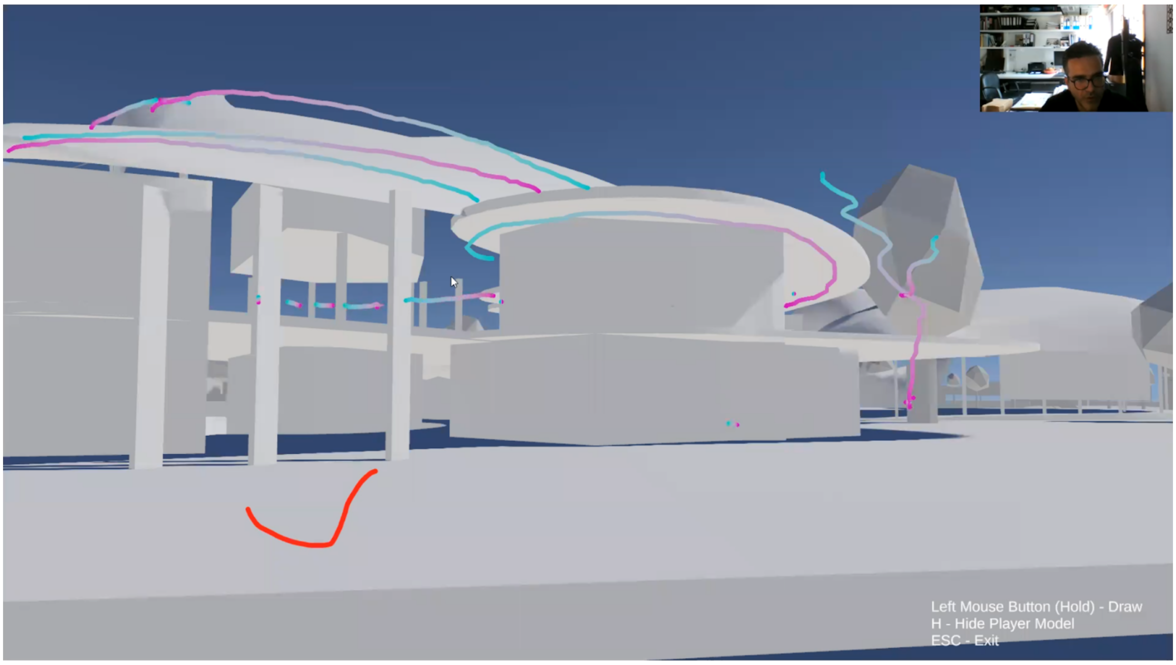

Figure 5.54. Using Airsketch to Give Criticism

In the following weeks, several groups of students chose to present their work in OVS according to their own choice, without any influence from the tutors. These groups of students tend to do walkthroughs inside their projects, in addition to the physical material they brought. As an interesting approach, one group of students did the walkthrough while pointing to the location of the avatar on the printed plans. These instances were regarded as pointing towards a potential in using such environments even in a face-to-face setting, which implies the possibility of a hybrid design studio.

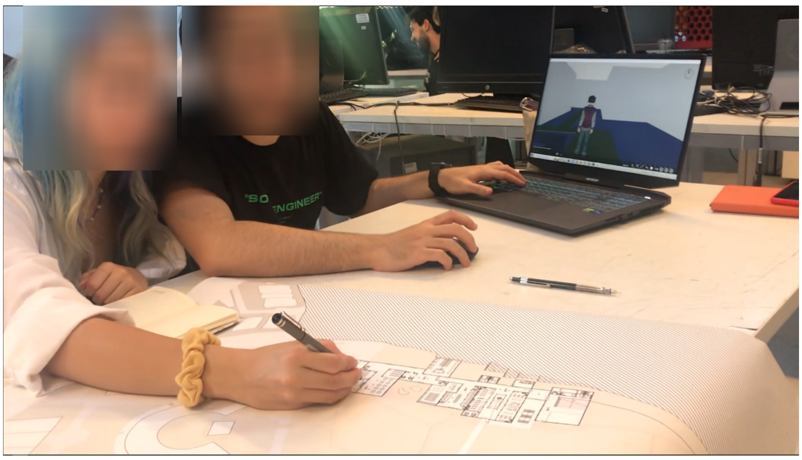

Figure 5.55. Physical Media and OVS

On another subject on the use of virtual environments in design studios, the same group that did walkthroughs while pointing on the physical plan, also integrated color-coded walking paths in their design. These paths were utilized for wayfinding for certain scenarios in their project, and were also experienced and discussed in OVS. It was observed that as students became more acquainted with OVS and used it more often, their decisions leaned towards more experience-oriented designs. In a similar approach, another group of students often demonstrated the vistas in their project through using OVS.

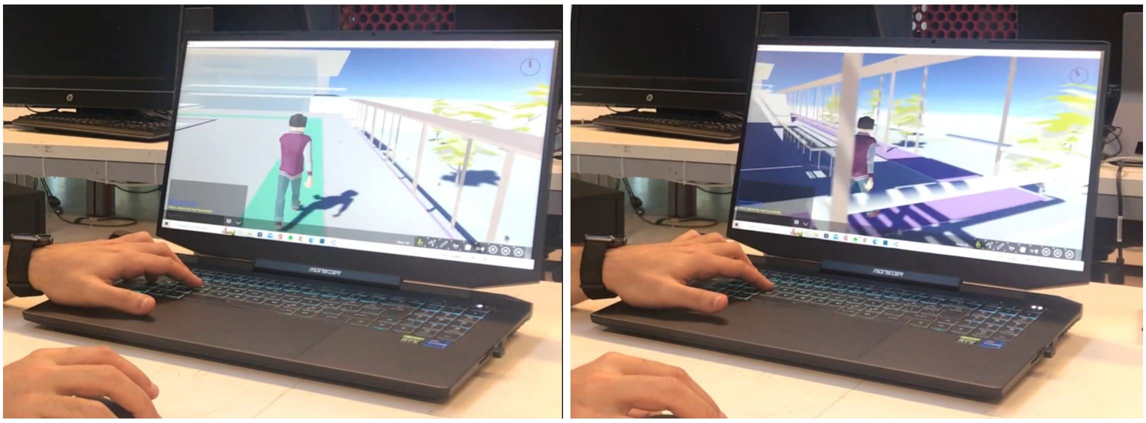

Figure 5.56. Color-Coded Paths

5.7. Feedback Survey

A survey was prepared and attached to OVS **after the third design studio tutor interview**, which was initiated when quitting the software. The students and the tutors were asked to participate in it after each use, to collect usage data, recommendations and bug reports. However, the participation was quite lower than expected and did not produce any meaningful results or insight. Due to this, no further analysis was made on the responses and another form of analyzing usage had to be done, in the form of a "mock jury", entirely in OVS.

The questions were aimed at collecting usage statistics, requests and error reports. OVS itself was and still is not equipped with such analytical methods to collect usage data yet. It was left up to the users to share such information if they chose to do so. The students of the Architectural Design Studio led by Tuğrul Yazar and Nilüfer Kozikoğlu were often reminded to fill out the feedback form after each use, which wasn't kept mandatory at any time.

On the topic of integrating such usage statistics and the related methods into the software, it is possible to implement counters and timers to certain features and interactions. Recording and storing this data is found to be fairly easy, although transmitting this data requires a dedicated online storage or a database, which wasn't developed during the timeframe of this research. Instead, additionally providing a chance to ask for personal opinions on certain subjects, a survey which was automatically launched when exiting the software was found more suitable for the time being.

It should be noted that during the process of the preliminary study survey and the feedback survey, it was found rather difficult to have students participate in studies such as surveys. It would perhaps be more efficient to prioritize the aforementioned automatic usage data collection methods in future studies.

The survey was prepared in Google Forms, and the responses were collected anonymously with no restrictions on submission amounts as the users were expected to fill out the form after each use. Below are the questions asked and the collected responses.

Question 1:
What was the build version (date) of OVS?

This question was included to associate the issues and the feature requests with the version of the software being used. It was found that without an automatic updater/patcher, students often tend to not check for newer versions of the software. A repository with the most recent version of the software, along with announcements was shared with the students, although it wasn't checked regularly, and the students had to be reminded to update the software in class sessions in which OVS was used.

As the version logs and the differences are not of importance regarding this research, the responses of this question will be omitted. Very few features were implemented during this term and the development was mostly focused on fixing bugs, server stability and QoL (Quality of Life) improvements.

Very few responses were collected throughout the term and especially during the initial weeks of the studio. As a result, even though there were some feature requests and certain issues spotted later, the remaining time for such improvements was found to be insufficient.

Due to the fact that most of the responses were collected in the final weeks, the answers to this question was insignificant even from a development perspective as most of the respondents had used the same, final version of OVS.

Question 2:

Approximately, how many times have you used OVS?

This question was asked to provide insight on the respondent's experience with OVS, regarding the following questions.

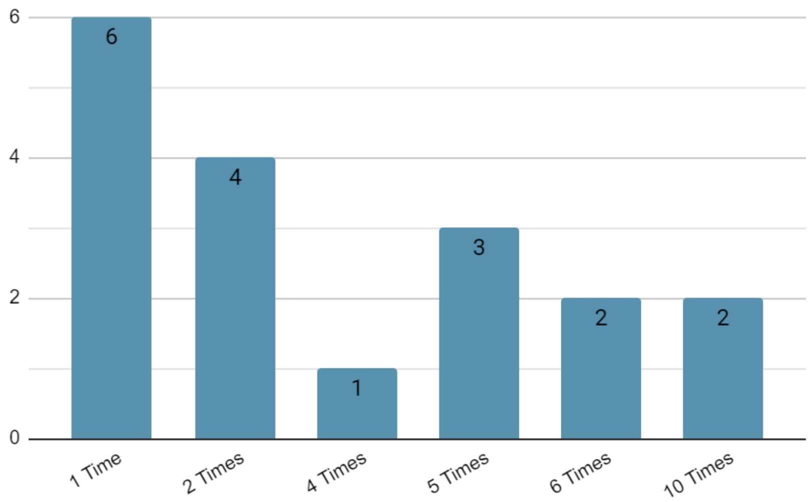

Figure 5.57. Answers for Question 2

Answers for question 2:

- 1 Time 6
- 2 Times 4
- 4 Times 1
- 5 Times 3
- 6 Times 2
- 10 Times 2

Although submitted anonymously, the responses can be tied to the submission instance, which provides insights on the issues faced, through the usage amount of the respondent.

Question 3:

Who are you?

Although the question may seem blunt, it was asked to determine the role/level of the respondent in the studio. This was a single-choice question.

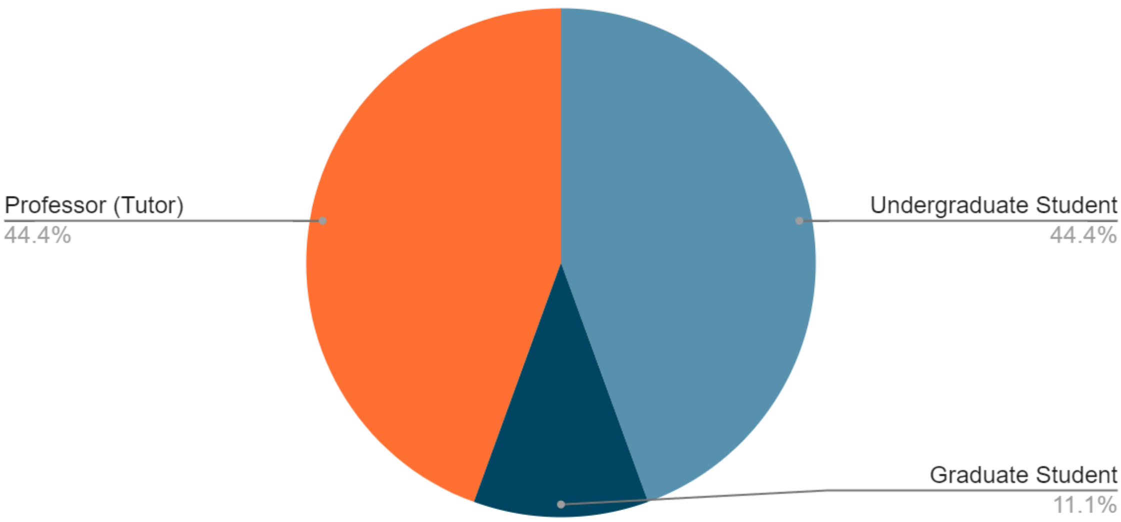

Figure 5.58. Answers for Question 3

Answers for question 3:

- Undergraduate Student 8
- Graduate Student 2
- Professor (Tutor) 8

It is apparent that the tutors were more participatory towards the survey, in regards to their ratio in the class. When the rest of the questions were analyzed and matched with the profiles, the survey tends to project answers from both sides of the communication, equally. OVS and some information regarding the research was also shared with a group of graduate students from Istanbul Bilgi University, thus providing insight from them as well through the feedback survey.

Question 4:
What was your role in the network?

Having some technical issues related to the software, addressed as “bugs” in software development jargon, it is important to understand the role of the users in terms of networking in a multi-user environment. As the code regarding the host, clients and the server are written separately, knowing the role of the user helps with identifying bugs and other issues. As a result of the server executables not being shared with the students, the only roles that the participants could have been in were host and guest (client). This was a single-choice question.

Answers for question 4:

- Host 4
- Guest 14

Question 5:
How many people were in the session?

In order to determine usage trends and issues in various user capacities in sessions, this question was asked to link some of the following responses to the session sizes. This question was asked with a text box as it was foreseen that the participants might not remember the exact number of users, or the user amounts may change during the sessions.

Answers for question 5:

- 1 User 4
- 2 Users 1
- 3 Users 2
- 4 Users 1
- 5 Users 1
- 6 Users 3
- 7 Users 1
- 8 Users 1
- 10 Users 1
- 12 Users 1
- Approx. 5-8 Users 1
- Approx. 10-15 Users 1

Question 6:

What was the purpose of your usage?

This question was asked to determine the usage scenario per participant to be analyzed further with their remaining responses. It was a multiple-choice question.

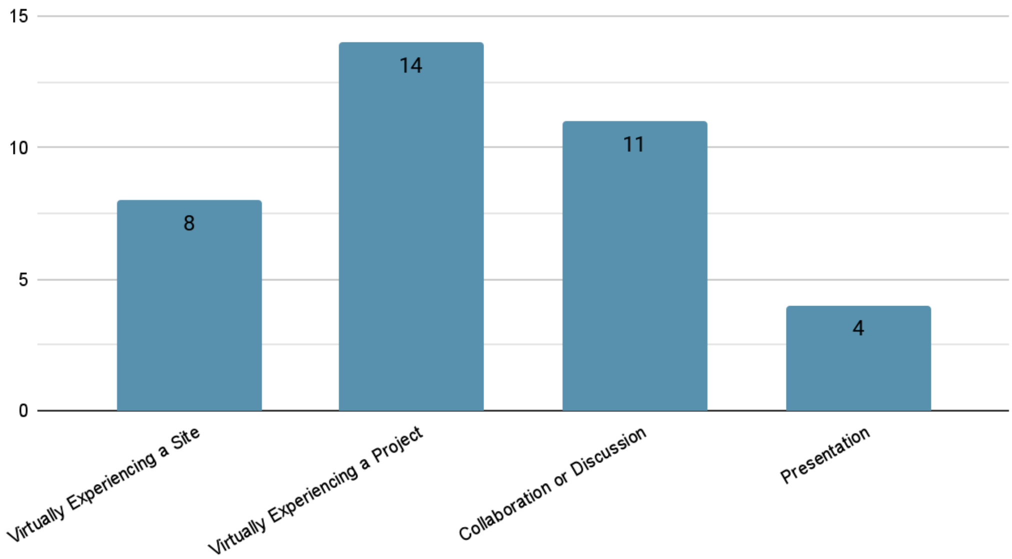

Figure 5.59. Answers for Question 6

Answers for question 6:

- Virtually Experiencing a Site 8
- Virtually Experiencing a Project 14
- Collaboration or Discussion 11
- Presentation 4

These responses could only be used to provide insight on the answers to the other questions in the survey, not in a way that could provide insight on OVS's usage preferences. This is due to the fact that all of the students in the studio had used OVS to explore the project site, present their work and discuss each other's project during the semester. The numbers emerging from the responses do not reflect the usage trend on the possible scenarios, thus could only be used in combination with other responses to determine correlations between the usage scenarios and other issues and tendencies.

Question 7:

If the session was associated with your project, in which phase do you consider your design to be in?

This question was asked in order to determine usage trends regarding the stage of the project that is being discussed inside the environment. As in, whether users tend to use certain tools more often in certain stages of design. This was a single choice question with a scale from 1 to 5, signifying the stage that the design is from "Early" to "Late".

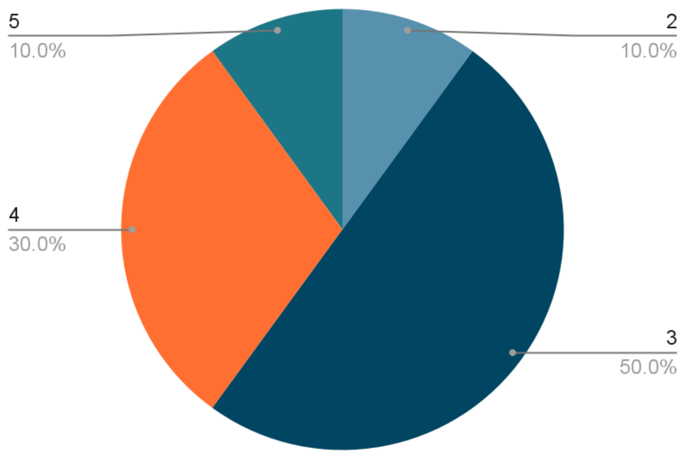

Figure 5.60. Answers for Question 7

Answers for question 7:

- 1 (Early) 0
- 2 1
- 3 5
- 4 3
- 5 (Late) 1

Question 8:

With whom were you in a session with?

This question was asked to determine the setting in which the session was held. The various combinations of user profiles could have implications on which tools are often used between types of users. This was a multiple-choice question.

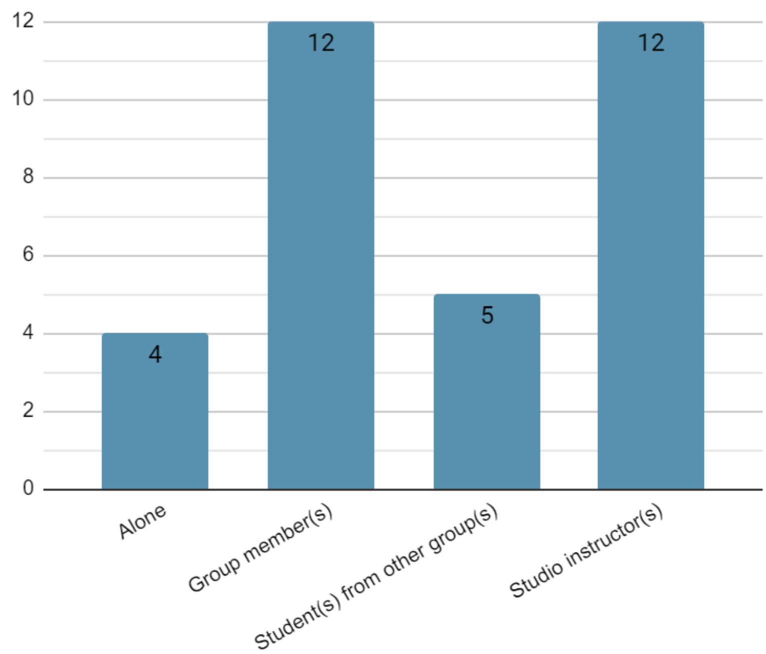

Figure 5.61. Answers for Question 8

Answers for question 8:

- Alone 4
- Group member(s) 12
- Student(s) from other group(s) 5
- Studio instructor(s) 12

Question 9:

Which of the following features have you used?

This was asked in order to determine inclination towards certain types of features in specific scenarios, which would be linked through the answers for previous questions. It was also aimed at uncovering tool and feature preferences in OVS, and to determine if certain features were preferred more than others, or perhaps used at all. This was a multiple-choice question.

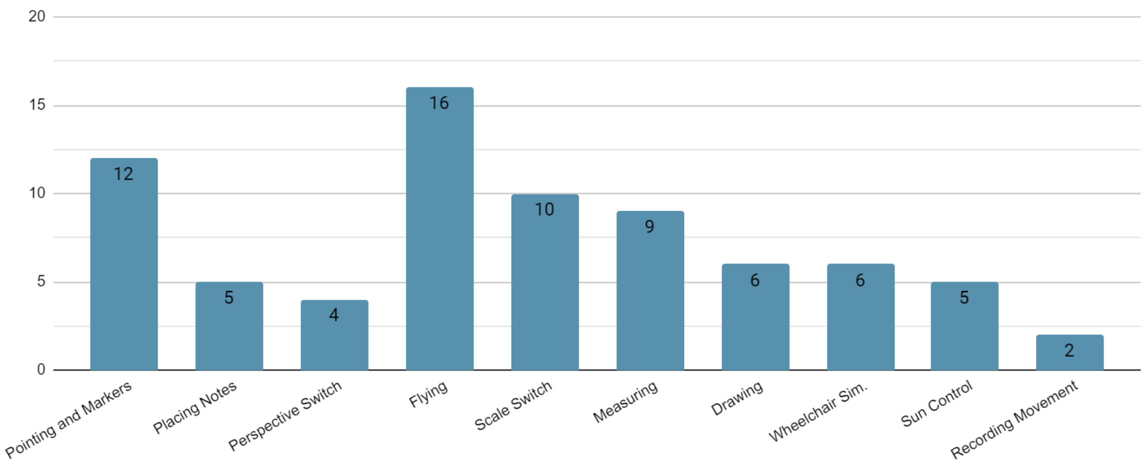

Figure 5.62. Answers for Question 9

Answers for question 9:

Pointing and Markers	12
Placing Note	5
Perspective Switch	4
Flying	16
Scale Switch	10
Measuring	9
Drawing	6
Wheelchair Simulation	6
Rain Simulation	0
Sun Control	5
Recording Movement	2

From an initial analysis, it was determined that the simulation and analysis methods in OVS were preferred less than traversal ones, which was also observed in studio sessions. As the participation was very limited, further analyses tying certain scenarios to usage preferences were not made.

Question 10:

How would you rate the experiential aspects of OVS?

Concerning the experiential aspects, as in experiencing their designs or the environment in real-time, this question was asked to determine if participants had found OVS to be preferable in such cases. This question was asked in the form of a scale from 1 to 5, corresponding to “Very Bad” to “Very Good”.

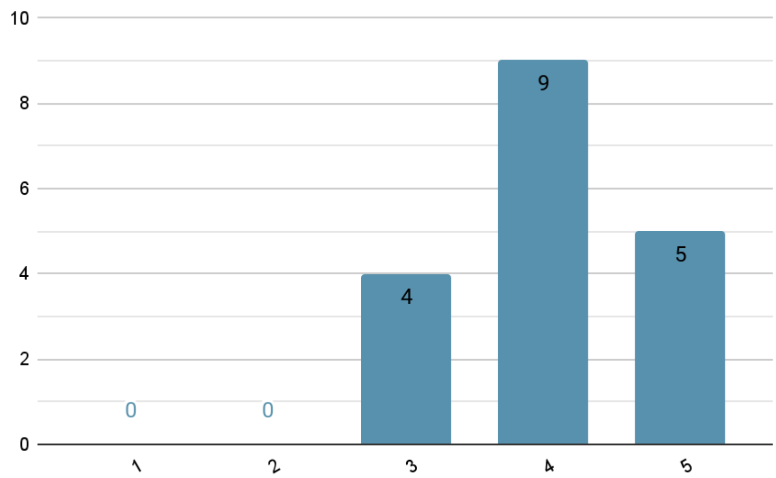

Figure 5.63. Answers for Question 10

Answers for question 10:

1 (Very Bad)	0
2	0
3	3
4	4
5 (Very Good)	5

Question 11:

How would you rate your communication inside OVS?

This question was asked to determine if the participants had found their communication inside OVS to be efficient; as in, if they had found themselves to be successful in expressing themselves inside OVS. It was asked in the form of a scale from 1 to 5, corresponding to “Very Bad” to “Very Good”.

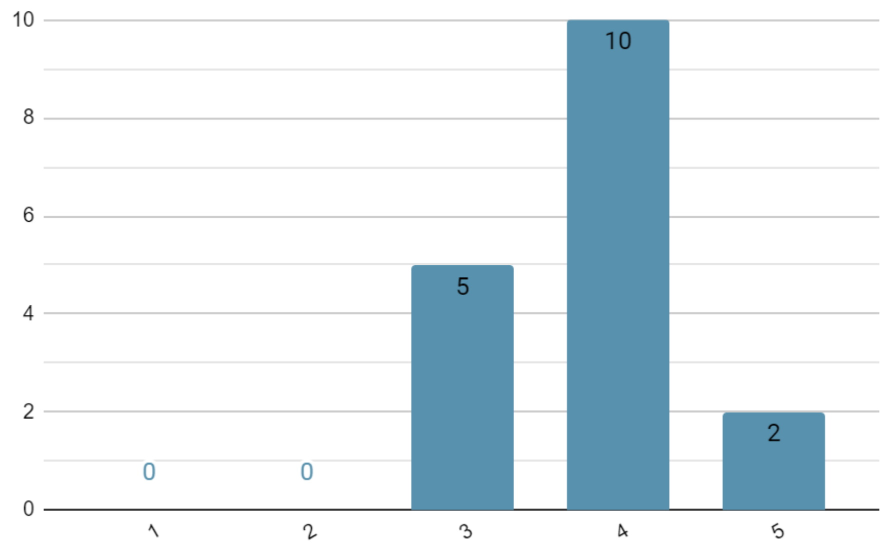

Figure 5.64. Answers for Question 11

Answers for question 11:

- 1 (Very Bad) 0
- 2 0
- 3 5
- 4 10
- 5 (Very Good) 2

Question 12:

How would you rate the multiplayer aspect of OVS in terms of communication?

This question was asked to determine if the participants had found the multi-user approach in communication to be useful; as in, if they had found a multi-user, real-time environment to be more efficient in communicating with each other. It was asked in the form of a scale from 1 to 5, corresponding to "Very Useless" to "Very Useful".

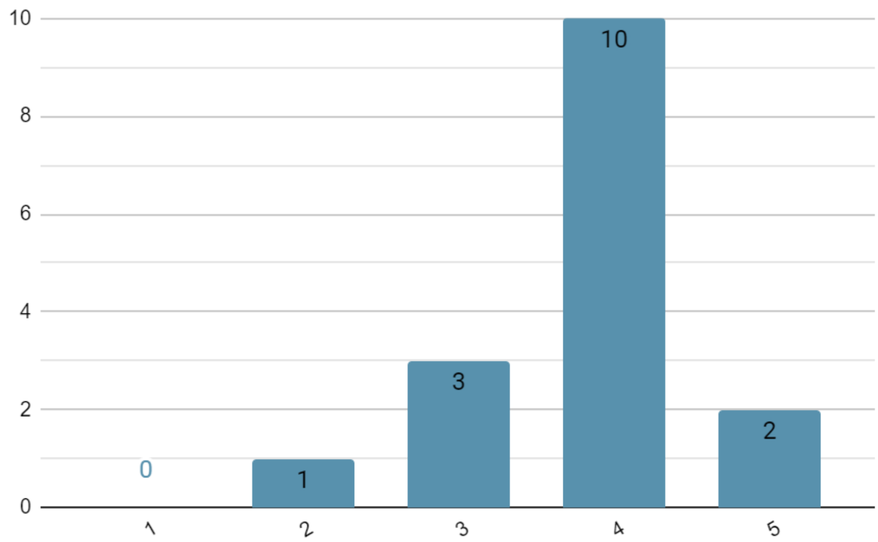

Figure 5.65. Answers for Question 12

Answers for question 12:

- 1 (Very Useless) 0
- 2 1
- 3 3
- 4 10
- 5 (Very Useful) 2

Question 13:

How would you rate the analysis/simulation aspects of OVS?

This question was asked to determine if participants had found the analysis and simulations inside OVS to be useful. It was asked in the form of a scale from 1 to 5, corresponding to "Very Useless" to "Very Useful".

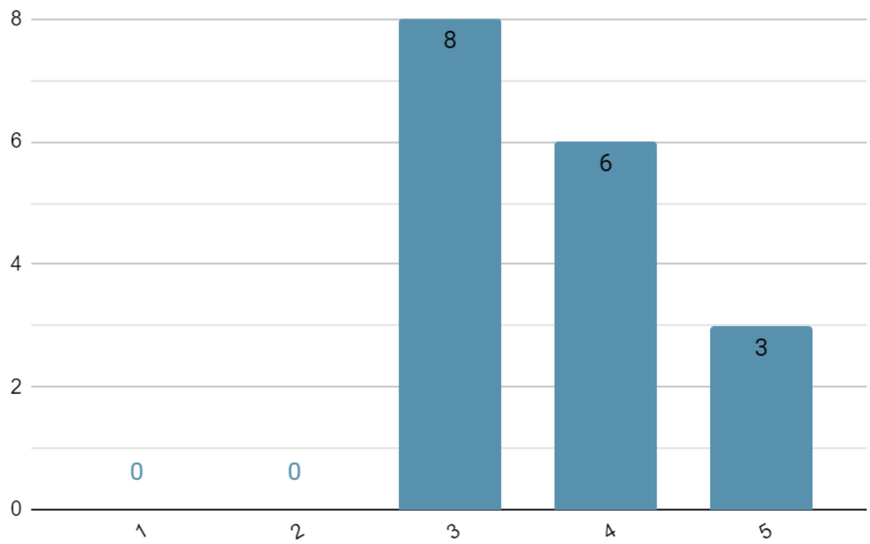

Figure 5.66. Answers for Question 13

Answers for question 13:

- 1 (Very Useless) 0
- 2 0
- 3 8
- 4 6
- 5 (Very Useful) 3

Question 14:

Did you realize anything new about the site or your design while you were in OVS?

After the initial tests and the interviews, it was observed that traversing through a virtual environment either through a first-person perspective or a third-person one, users often discovered aspects that they haven't been able to discover or realize before. As an example, during the very first tests which were neither recorded nor previously discussed in this research, participating students expressed that when they experienced their design in OVS, they had the opportunity to realize that the height of their railings were too high compared to what they have perceived them as while designing in a CAD environment, and that their spaces were too sunken in terms of volume and ceiling heights. In order to examine if similar opportunities could be found, either tied to traversal methods and scale perception, or an entirely different aspect of OVS, this question was asked with a follow-up question to determine if this environment could provide certain revelations and discoveries for the users. This first part of the question was a single-choice query.

Answers for question 14:

- Yes 9
- No 9

Question 15:

Please explain if you've answered yes to the previous question.

This follow-up question took responses in a text box format, and was thought to provide insight on the previously mentioned possibilities.

Answers for question 15:

- "Scale; spatial relationships and interior spaces are more understandable"

Possibly referring to using a humanoid avatar, this respondent expresses that the interior spaces and spatial qualities became easier to understand when perceived from a realistic perspective, with a scaled figure.

- "The scale differences, the angle of the ramps (it was too high for wheelchairs), vistas from the interior"

Aside from referring to better perception of scale, this respondent also mentions that they had the chance to understand that their ramps were not fit for wheelchair usage. They also explain that they had the chance to examine the views from inside their design through using the included camera and traversal methods.

- “Mostly, we have examined that we have used too long paths. We walked our design in a more realistic sense. Then we realized that some parts were too long to walk and some were too short.”

Through having a walkthrough of their project, this group realized that some distances were longer or shorter than they had initially perceived while designing.

- “The scale of a human with contrast to the project site”

This respondent seems to indicate that they have better realized the scale of the project site through the use of a humanoid avatar.

- “Perceiving the immediate surroundings from inside the building was new.”

Similar to having the chance to experience the interior of their projects from the human scale, this respondent seems to state that they also had the chance to experience the surrounding environment from inside their project. This could be tied to the fact that the students were expected to model their projects including at least the immediate surroundings.

- “It raised awareness about scale in some areas of the project.”

Again, commenting on the scale perception, this respondent expresses that they have increased their awareness of scale in their project.

Perhaps due to the fact that the response outcome was quite low, almost all of the responses and the discoveries made were tied to avatar usage and perspective modes. It was thought that if the response count was higher, this could bring more variation and some other opportunities of this environment could have been discussed.

Question 16:

How would you rate your overall experience in OVS?

This question was asked to trace the reasons behind lower satisfaction in respondents. Lower scoring responses would be investigated further through their answers for the rest of the questions. It was asked on a scale from 1 to 5, corresponding to “Very Bad” to “Very Good”.

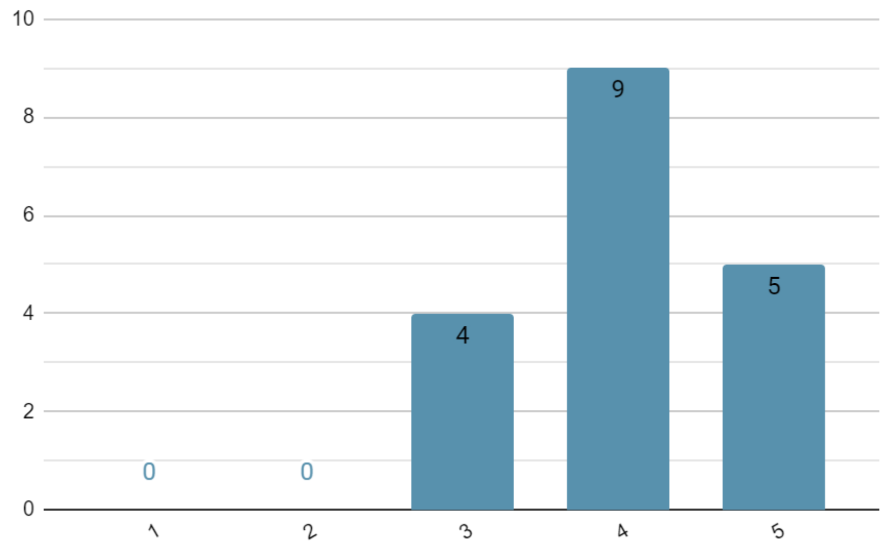

Figure 5.67. Answers for Question 16

Answers for question 16:

- 1 (Very Bad) 0
- 2 0
- 3 4
- 4 9
- 5 (Very Good) 5

The rest of the questions in the survey were related to bugs, errors and features requests. Aside from the requests, as the responses to these questions are not of importance in the scope of this research, they will be omitted. Although, the questions will be included and explained to provide reference for further studies.

Question 17:
Do you have any recommendations or feature requests?

This question was asked in order to prioritize the development of certain tools or add entirely new ones to the feature list if found necessary.

Answers for question 17:

- "It may be nice if we could add lights, sunlight is good and shadows are also good but sometimes it is hard to experience dark areas and some areas can be overexposed"
- "guided touring"
- "Maybe if we could join with more people."
- "Some parts of the model could be interactive, providing details or more information when clicked, adding an extra layer of detail / avatars could be a bit less intrusive or more abstract / background could be less intrusive"
- "There are some bugs, fixing them can further improve the use of the program."
- "A simpler design may be preferred for the background and avatars. While the student is describing the project, a tracking sign can be used so that we can more easily follow the point she/he has explained."

Question 18:
Did you experience any connection issues?

This question was asked to determine the percentage of connection issues.

Question 19:

Could you explain the connection issues you've encountered? What happened, what might have caused it?

As a follow-up question to the previous one, this question was asked to determine the reasons behind the connection issues. Detailed explanations can help with re-enacting the issues, alas finding the bugs in the code and fixing them.

Question 20:

Did you import any models?

This question was asked to determine if in certain scenarios, some roles were unexpectedly importing models into the environment, and combined with the related previous questions, how frequently.

Question 21:

If you have imported models have you experienced issues? What format and software did you use to export your model? Please explain in detail what the issues were.

Similar to questions 18 and 19, this was asked in order to find and fix bugs in the importing pipeline, as well as the CAD software usage trends among participants, and if some issues were related to the used CAD software.

Question 21:

Please upload images of the importing issues here

As some issues in 3D models are better shown than explained, an image submission box was included after the previous question.

Question 22:

Did you experience any bugs except above?

Keeping the pattern on extracting percentages, this question was asked with a follow-up question for elaboration. Connection and importing issues being the most common and critical ones, other bugs were not prioritized above them unless found more critical, such as the ones incapacitating users.

Question 23:

Could you explain the bugs that you've encountered? What happened, what might have caused it?

As with most of the bugs and issues, more details can help re-enact them.

Question 24:

Error logs

This was an upload box for error logs that Unity Crash Handler produces on unexpected termination. Combined with a developer build and verbose logging, such error logs can point to the exact line of code that caused an issue.

5.8. Online Virtual Jury

Following the unsatisfactory outcome of the feedback survey, three student groups from the architectural design studio section in which OVS was tested throughout the semester were picked to repeat their final presentations to a separate group of tutors. However, this presentation/jury would be done **entirely in OVS**. This session was recorded and analyzed to provide insight on a real-life scenario on the usage of a *multi-user virtual environment in architecture*. The chosen students were comfortable in using OVS, as well as some of the tutors who participated, as they were also present in the previous interviews. The remaining tutors were given instructions on how to use OVS before the session. All the participants were asked to not talk about OVS during the session but rather focus on the presentations themselves. The online virtual jury was conducted on 04/07/2022 and lasted 135 minutes.

Prior to the presentation, the groups were asked to design a holistic jury presentation environment with their models built-in, rather than importing their work during the presentation. A discussion was held and a settlement was reached. The environment was designed to be four pieces of rocks floating in the air, with circular platforms attached to the top of the rocks. The skybox was chosen to reflect a floating environment among clouds, and the skybox was programmed to rotate, giving the environment a dynamic feeling. The arrangement of the space was done in a triangular formation, with one of the platforms in the middle. The platform in the middle would be the spawning point for the participants, with the studio brief hung onto display boards on it. The other platforms were designated to each of the student groups and were connected through bridges, with circular stands at the center of them, which held their 3D models. The models were surrounded by display boards that held the presentation posters. It was decided that the bridges would follow a linear path, rather than each of the platforms being connected to each other. The presentation scenario was decided as starting with the presentation of the material on the boards, seeing the scaled models from the outside, and then hopping on them to be microscaled. When the participants would be microscaled, the models would be explored in 1:1 scale and then the discussions would be held in this scale, with the opportunity to go back to the presentation boards.

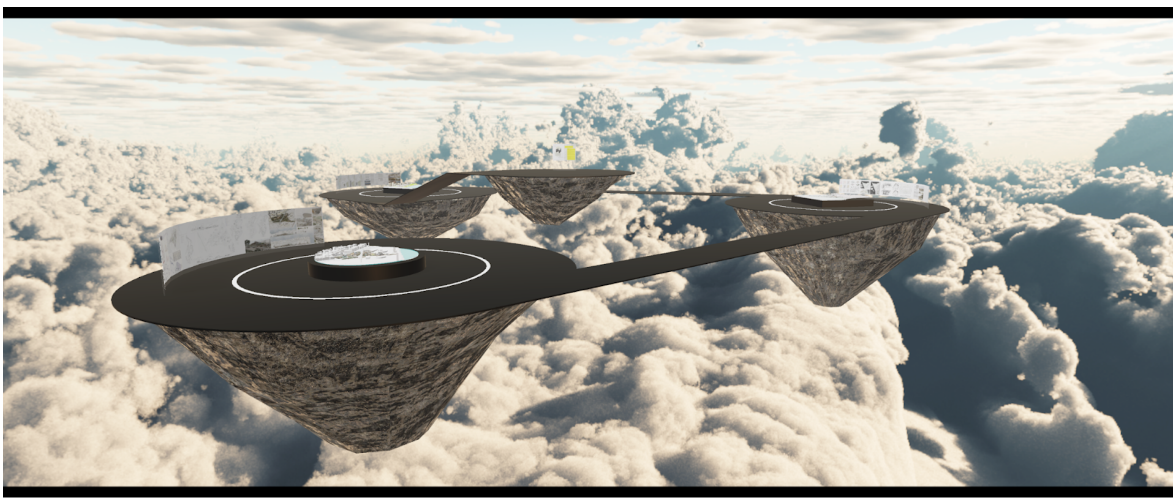

Figure 5.68. The Presentation Environment

The online virtual jury was held through this special build of OVS, with all of the students and tutors connecting to a dedicated session hosted through the server in Istanbul Bilgi University. A teleconference session was held in parallel to provide voice communication.

Figure 5.69. A Model Display

At the beginning of the session, the participating jury members were again explained the aim of this meeting and were asked to contribute in this simulation of a jury through only engaging in the discussions themselves, rather than commenting on the software or the proposed method, until the jury was over.

As the researcher, and the developer of the software, no comments or directions were given past this point, until the end.

The jury session started with the jury members viewing the project brief on the display wall and the studio tutors Tuğrul Yazar and Nilüfer Kozikoğlu explaining the studio subject, the context and the surrounding area. Parallel projection cameras and zooming on first person perspective was utilized to read the text on the poster.

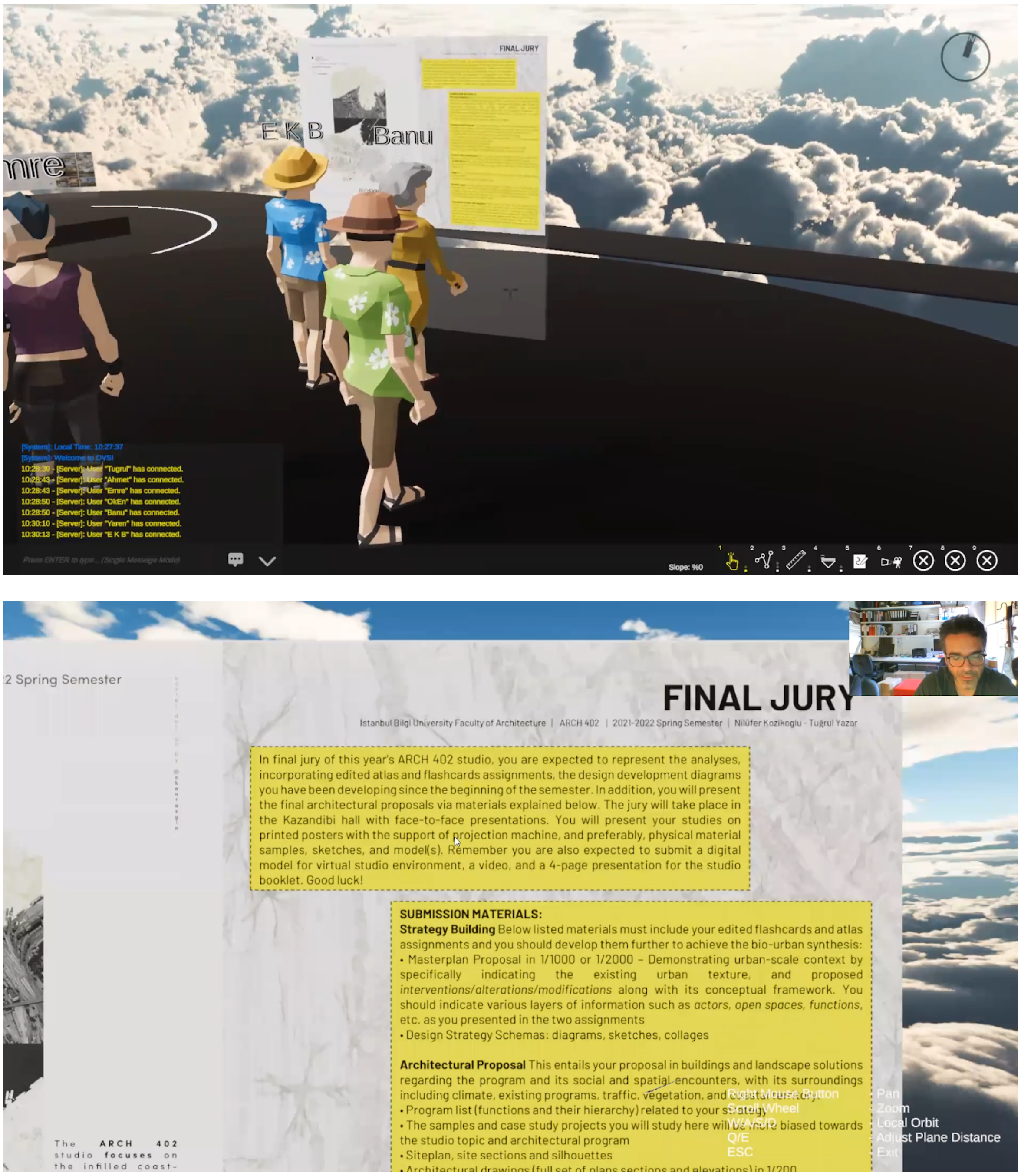

Figure 5.70. (a) Tutors Looking at the Brief
(b) Viewing the Poster in Parallel Projection Camera

After the introduction, the first group of students asked everyone to follow them to the first presentation island. Name tags were utilized to identify the students at this point. After the students introduced themselves, as they asked the jury members to follow, the jury members could only do so by identifying them through their name tags.

Even though the environment is an unconventional one, as a result of the presentation setup being done in a conventional way, the students started by presenting their posters as they would in a conventional, face-to-face environment.

Figure 5.71. Student Presenting Their Work on Posters

Pointing and marker usage was commonly utilized and became natural after the first few minutes. Both the students and the tutors were constantly pointing and marking when talking about certain things.

The advantages of having individual perspectives was felt immediately, the jury members were seen looking at other parts of the posters and the 3D models individually as the students were explaining their projects. However, as some tutors became too close to posters or walked in front of others, they obstructed the view of others and were unaware of this situation. A need for avatar transparency or turning off the visibility of avatars entirely was found necessary at this point, as it was also observed that some participants were getting distracted by the other participants' avatars.

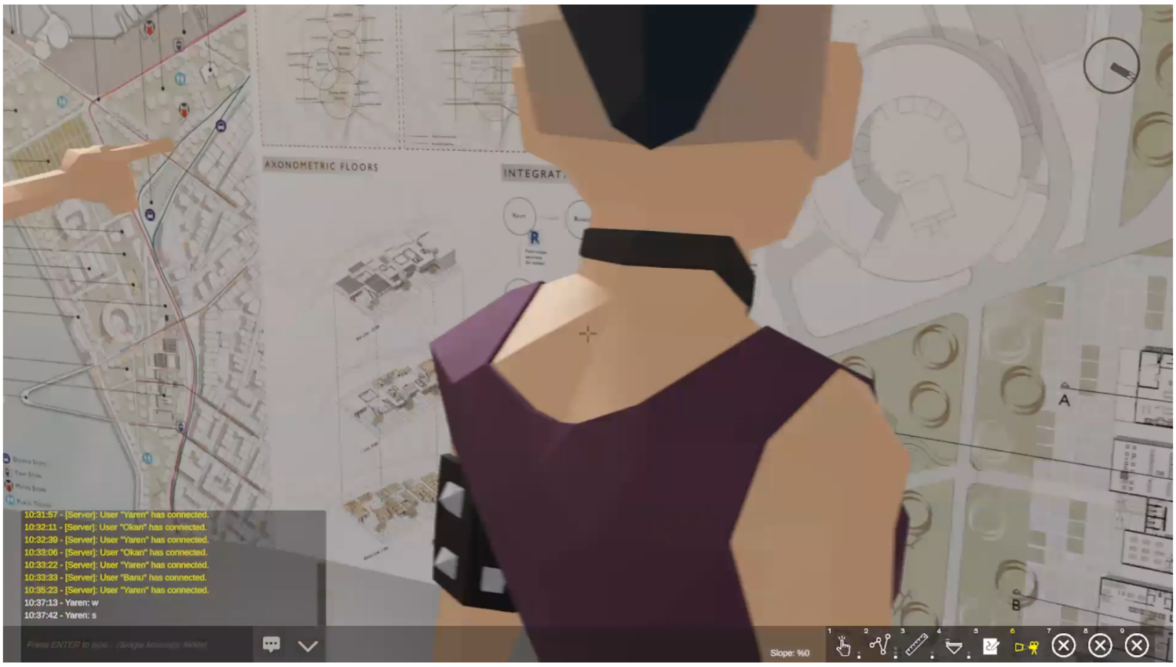

Figure 5.72. Visual Obstruction by Avatars

This distraction was also caused by an unintentional consequence of this game-like environment. The participants were less serious than they would be in a face-to-face environment and were often doing or trying distracting things such as jumping over objects or eachother, flying around or dancing.

The visual obstruction caused by avatars was soon realized and the participants started to act as they would in a physical environment, by moving away from the board to let everyone see. At this point, the utilization of pointing and zooming also decreased and the avatars seemed less responsive, which indicates that the participants were utilizing the parallel projection camera instead. As the pointing and zooming approach was decreased, so did the workspace awareness between participants. Although letting some obstruction occur, this method would also let the participants be aware of each other's view. It was apparent at this point that even when the users are detached from their avatars, there should be a way to keep track of their points of view.

Figure 5.73. Participants Utilizing Avatars and Pointing

Aside from the simulated conventional way of presenting an architectural project in a physical studio, the participants later moved into the 3D model itself, exploring the project in 1:1 scale. The participants instantly employed flying controls in this scale, and the first signs of the lack of a map or a minimap were found.

Figure 5.74. Participants Flying Inside the Model of a Project

The presenting students asked the jury members to not fly too far away to avoid getting lost and have them meet near a landmark in their model so that they can continue with their presentation. The landmark that they have chosen was a pavilion inside the park attached to their project.

Figure 5.75. Participants Meeting in the Park

After an initial fly-through and meeting, avatar scaling was reminded to the participants as one participant scaled themselves, and it was utilized to get a more general idea of the project through a wider perspective.

After having the site explored and seeing the project from outside, students led the jury members into the building. In this scale, participants initially utilized first person perspective camera mode along with walking to navigate inside the places. However when they felt lost or stuck, they switched to flying to get out of the interiors, find others and reorient themselves.

The disorientation and getting lost was found mostly to be due to the lack of detail in the interiors of the model. As the walls and floor became monochromous and the rooms became big gray boxes with few openings, distinctions between spaces could not be made and the participants felt the urge to go outside to reorient themselves.

Figure 5.76. Supersized Avatars

After the presentation and the guided walkthrough, students asked the jury members to experience the project by themselves, which wasn't discussed priorly. This was found to be a very similar approach to a touristic excursion. The participants took some time to explore the project on this scale and called the presenters out when they needed further explanation on certain parts of the project. This also proved to be an opportunity in discussing parts of a project which aren't existent on static presentation material such as posters.

Exploring the project in a human scale was found particularly useful in the interiors as certain ergonomic and volumetric qualities were identified easier through the use of avatars. The students further commented that they have felt like real estate agents or developers while they were leading the group into the interiors while explaining the functions and spatial qualities of the rooms.

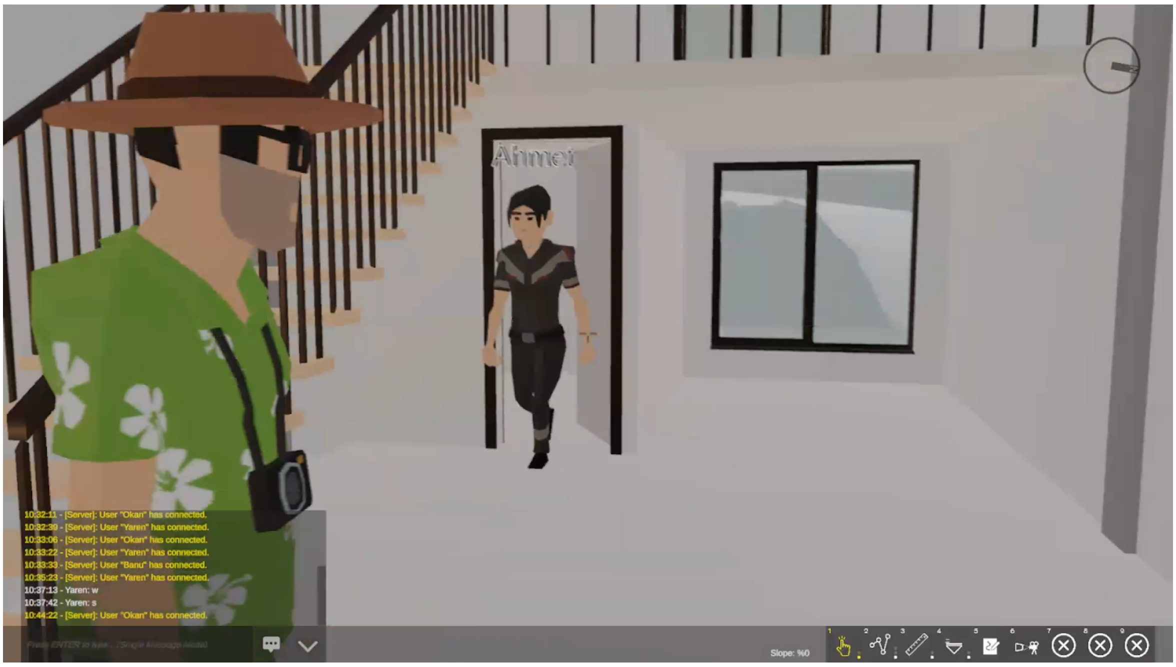

Figure 5.77. Exploring the Interiors With a Reference to Human Scale

After the first group's presentation, during the discussion stage, the participants who were first-time users stated that they couldn't follow the presentation of this first group as they were still trying to get used to the environment and the controls. This was overcome by the time the second group was presenting and the same participants stated that spending just an hour or two in this environment would make them comfortable in using it, and that the learning curve wasn't steep at all.

The participants who were used to the environment however, utilized their avatars and annotation tools to discuss the project in this period. Pointing, temporary markers and 3D lines were the most utilized features. It was expected that airsketching would be the most utilized tool by the jury members. However, on the contrary to being the most requested feature by the studio tutors during the interviews, it wasn't used during this entire jury session.

Even though the models were developed to a certain level of detail, the lack of cluttering objects and environmental factors such as human figures were felt. As the students tend to add such details on post-processing, the 3D model itself was rather bland and unlively compared to the prepared static visuals. This brought the necessity of a prefab object library along with NPCs for such cases into question, where users could even spawn some objects while presenting. In more crowded sessions, the users themselves would fill the need for human figures and could even act like potential visitors, though in this instance, the session consisted of fewer participants and due to the scale of the project, the spaces felt rather empty.

It was observed that the users who wished to see the project from a larger scale would start clipping with the environment to achieve certain perspectives. This again brought the issue of obstructing the view of other users, thus the necessity for optional avatar transparency. However if a more detached approach was utilized, such as a pan and orbit camera implementation, while negatively affecting workspace awareness, this approach could eliminate this necessity.

Figure 5.78. Flying Supersized Avatars Clipping Through the Ground

The topic of disorientation was brought up many times and it became apparent that the development of a minimap should be prioritized and implemented as soon as possible. Due to participants getting lost or becoming disoriented, the participants had to regroup often.

Figure 5.79. Participants Moving On to the Next Project

The designed presentation setting was found to be useful in countering the often experienced disorientation through having a linear path for the presentation order. The jury members could always track which project was being discussed due to this setup.

An observation was made on the students' behavior in OVS; where they thought that they couldn't express their design intentions properly or miss some aspects in trying to describe their design through the static presentation material that are on the posters, they often tend to lead jury members into the 3D model to display such aspects or qualities.

Moving on to the criticism of the online virtual jury and OVS, numerous comments were received both in positive and negative terms. Overall, most of the comments should be analyzed while keeping in mind that some participants were struggling with getting used to the environment as it was their first time using it.

Participating jury members commented that having the opportunity to view the posters and the projects from their own perspective with the ability to zoom in on details was extremely useful compared to conventionally used methods in distance education.

On the other hand, they have also found having to navigate the environment while trying to follow a presentation to be exhausting. They suggested that an observer/spectator mode would be useful in such cases, where there's a presenter explaining specific things.

Regarding markers, some participants stated that it took a while for them to realize that the markers were being utilized and the utilized spheres weren't enough to draw attention. This was explained further by describing that the transparent visuals of the markers were making them hard to notice and strictly visual cues were not enough to draw their attention.

The opportunity to have 1:1 walkthroughs inside the projects along with scaling opportunities was found remarkably positive by the jury members. They stated that it helped them understand and experience the projects to a degree that was surprising to discover.

Having walked inside the projects in a human scale, the participants expressed that the issues that are faced regarding scale perception among students can be overcome through the use of the approaches employed by this environment. They stated that from being able to experience the projects in a human scale, they could more easily understand the real size of the projects and more precisely determine if certain design choices were actually appropriate for the human scale. An example was given in this regard; a jury member watched another participant run across a bridge in one of the projects and through this observation, the size of the avatar compared to the span and the time it took for the avatar to go from one end to the other, they determined that the bridge was actually longer than it was perceived in the 2D drawings and not as proportionate and humane as it may seem.

In an interesting take, it was explained that being in a more interactive environment brought the idea of interactable models. One jury member asked if a feature enabling model interactions could be implemented in a way that some elements in the project would be responsive or, similarly to permanent markers, store additional information such as system details.

On the topic of visual qualities regarding the environment itself, as in OVS, the moving clouds and the utilized avatars were found too distracting. The participants suggested that there should be options to alter these visuals individually, while not affecting other users. One participant suggested that the avatars should be more abstract and less detailed. This could be interpreted and implemented as client-side options for the skybox and avatar styles.

Adding to the previous discussion, certain underdeveloped models or technical issues can bring the issue of distracting visual quality. An instance of such issues happened in this session. As a common practice, students often assign distinctive materials of lower qualities to their models to later replace them in rendering software. One of the student models was developed in this sense and wasn't replaced prior to being imported into OVS, which produced an undesired visual quality in the environment. As discussed in the *Feature List* section, such issues can be overcome with shader programming, through replacing textures and materials in a stylized approach.

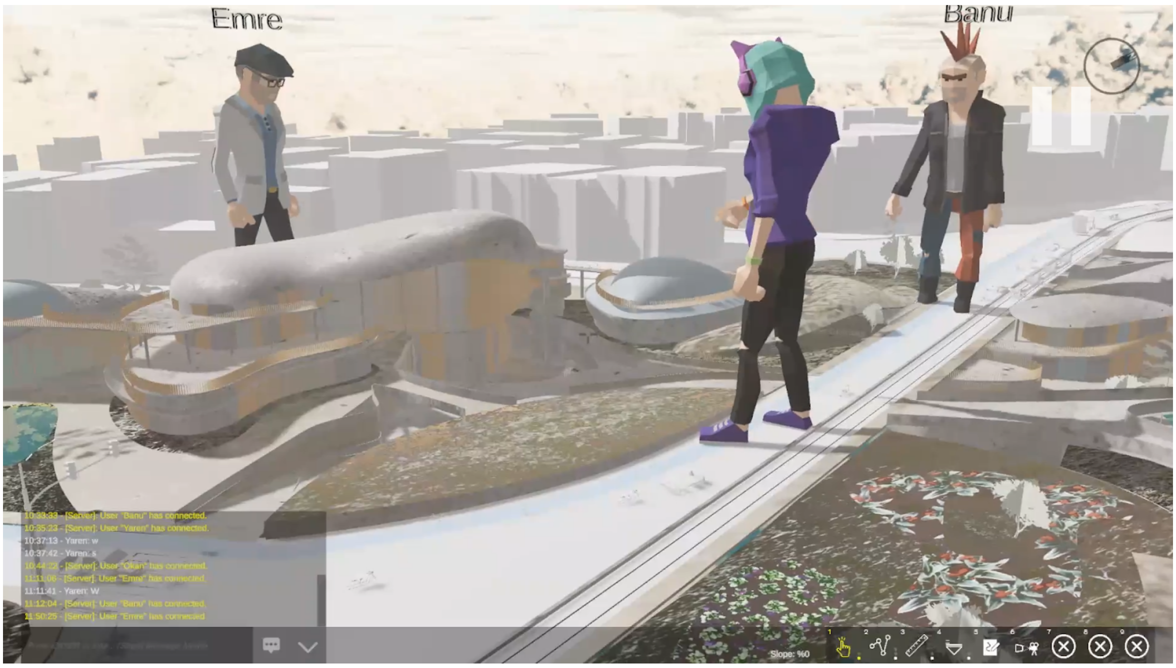

Figure 5.80. A Model With Placeholder Textures

On the scope of the research and the opportunities such environments can bring, a participating jury member stated that such environments, when permanent sessions are employed, can provide the means for international collaboration without time and distance constraints, as well as a virtual studio space for distance education and a possible hybrid setting. This was one of the assumptions of the research.

As a side note, the studio tutors commented that through using a medium in which every detail of the 3D models are reflected and easily inspected with no chance to hide errors, the students initially had difficulty in producing appropriate models. However, through these difficulties and in time, they have displayed progress and developed much better models in terms of 3D modeling, compared to previous terms and groups of students. This was interpreted both as a challenge in using such an environment, and a positive outcome.

6. CONCLUSIONS

In various stages throughout the course of the research, certain observations and deductions through the conducted studies were made regarding the use of multi-user virtual environments in architecture. However it should be noted that these results are tightly related to the chosen approach and the capabilities of the utilized software itself. As it was explained in the technical design, if another approach was selected and the requirements were interpreted in another way, these results could vary.

Overall, the described methodology was able to be followed. Regarding the final step, which is the feedback loop, the loop count was rather lower than expected. However, through conducting interviews and staging scenarios, the loops were more efficient than intended. The expectation of the feedback loops were that they would be brief and allow for rapid prototyping with smaller changes and numerous iterations. The reality however, was that these loops were limited in number, although providing much more insight than imagined, which allowed for bigger changes between iterations.

Regarding the research questions, below are the answers.

Q1. What are the challenges faced by traditional architectural design studios in the distance education model and what are the opportunities this model offers?

The aim of the preliminary studies was to answer this question. Through conducting the survey and the interviews, the challenges faced and the opportunities found were discussed and defined. Section 3.4, “Findings” (p. 83), is directly aimed at answering this question. However, it should be noted that the resulting findings could vary between institutes, and in the scope of this research, these findings are directly related to the education given in the Faculty of Architecture of Istanbul Bilgi University. If similarities between the requirements are found elsewhere, even if unrelated to architectural education, the following steps and features could be applicable, in theory.

Q2. For what purposes and to what extent can the potentials offered by virtual environments and game engine technologies be used in architecture, architectural education and distance education model?

Through the steps taken after the initial findings from the preliminary studies, the answers to this question were sought. By examining the literature and the existing software, the potentials offered were identified. After this examination, and through translating the requirements into features, the first part of this question was tried to be answered. Section 4, "Feature List" (p. 114), answers to the "for what purposes" part of this question.

The remaining part, "to what extent", however, had to be tested to be able to be answered. The steps taken in Section 5, "Digital Design Studio Case Study" (p. 146), were directly aimed for this part of the second research question.

The interviews and the sessions were recorded and analyzed to seek answers. The observations made were reported in the related sections. Numerous outcomes were deducted from these reports and will be explained below in combination with the answers to the third research question, as some of the outcomes are intertwined.

Q3. How can a digital/virtual architectural design studio contribute to architecture and architectural education?

The answers to this question lie in the observations made for the previous research question. Both as an advantage and a disadvantage for the research, the distance education period ended before the case study was conducted.

On the disadvantageous side, the participants were less inclined towards the use of a virtual environment as the studio sessions were conducted face-to-face. This resulted in the studies on capabilities and possibilities for distance education being impacted severely. Although due to unfortunate circumstances, there were some opportunities found to test such scenarios.

On the advantageous side, as most of the classes were conducted in a conventional face-to-face environment, the possibility of testing a hybridized environment was found.

The outcomes related to the second and third research questions (Q2 and Q3), both in positive and negative terms, are summarized below.

6.1. Findings

Even though limited in quantity, the sessions where participants connected remotely, much similar to the conditions of the distance education, posed as an opportunity to observe their behavior in such scenarios. In these sessions, the students were observed to be more participative as their presence was tied to their avatars, which required them to actively follow the sessions through controlling them. This was the intention behind the utilization of avatars. As this approach brought workspace awareness through additional features, it also forced the participants to be more active inside these virtual sessions.

Another outcome tied to this factor was the increased socialization between the participants, possibly as a result of the game-like environment. As the users were aware of the presence of others, they started communicating with each other, however due to the features related to communication not being fully developed, this was observed to be in the form of brief dialogues through actions such as rapidly spawning objects, social animations, and running or jumping around together, as their other methods of communication was limited through a single voice channel in the used teleconference software. Perhaps due to the environment leading the participants to communicate in this way, or the game-like environment itself, a decrease in seriousness was also observed. The students were often seen dancing, rather than listening to or joining in on the discussions.

Although the students were behaving in a less serious manner, due to the increased participation through presence, and the provided 3D annotation tools, the communication quality in this environment was found to be greater than the conventionally used distance education methods.

Aside from the presence and workspace awareness that was brought by utilizing avatars, an improvement on scale perception was achieved. As the participants experienced the environment with a human-scale avatar, they could more easily understand the distances and the spatial qualities of the site and their projects. This was brought to attention many times through the students criticizing themselves, as the proportions of spaces or objects felt off when they were experiencing their projects in the virtual environment.

As Bartosh and Philip (2019) and Moleta (2016, 2017) had explained in their research, an increase in experience-oriented design thinking was observed among the students. As their presentations were in the form of an experience inside the environment, they began designing with this form of presentation in mind. The students prepared presentation scenarios inside the environment and even marked routes for the walkthroughs of their projects.

Perhaps again indicating the effects of the experience-oriented medium, or a more technical reason, it was observed that the students paid more attention to materials and textures, and sun orientation in their designs. The possible technical reason behind it, is the concern for the visibility of their models. Due to the fact that flat and smooth surfaces pose issues in visibility and the geometries are harder identified in the absence of materials and textures, this may also be interpreted as an attempt to make their models more visible. Shadows are usually the only things that help with visibility in such cases, thus potentially indicating that sun orientations were utilized due to these issues.

Issues related to 3D modeling were observed to be recurring incidents throughout the whole term where the virtual environment was utilized. This posed both as a challenge and an opportunity for the students to improve their skills regarding modeling. It should be noted that utilizing the developed 3D virtual environment (OVS), regardless of the supported data types, required an environment to be modeled. Even though the senior architecture students in Istanbul Bilgi University are expected to be proficient in 3D modeling, this could have influenced students to modeling in cases where they aren't.

Even though the students were used to 3D modeling in this case, a lack of proper modeling etiquette was observed among the participants. The most common issues relating to this observation are the lack of attention to face orientations, inexperience and carelessness in converting model geometries and almost no experience in working with mesh models. As the students in Istanbul Bilgi University are used to working with Rhino as a 3D CAD software, they are often only familiar with NURBS geometries. Considering that game engines and software built with game engines work only with mesh geometries, knowledge and experience in converting models was required from the participants. Even though the participants were instructed and reminded of techniques regarding such conversions, a resistance was observed towards learning to operate with mesh geometries. As a result, the models either lost details, or were enormous in size due to students either choosing the lowest or the highest preset settings for automatic NURBS to mesh conversion.

Aside from the errors in conversion, the lack of details or unfinished models were easily detected in the environment. Post-processing rendered images is a common method of hiding errors among students, which is not possible in an environment where the model is experienced directly. Due to this fact, an increase in detailed modeling among students was observed as they experienced this situation. Very detailed models were brought by the end of term, which posed another issue, large file sizes and long importing times. Although the issues about modeling persisted throughout the semester, it was observed that students improved their modeling skills during this period.

Although not very common, issues in adaptation to the environment were observed. This could be due to the lack of alternative control schemes as other control patterns such as observer and spectator modes weren't developed. The participants who hadn't played video games utilizing the WASD scheme found it difficult to navigate in the environment, but most of the users adapted to the controls in a matter of minutes. This issue was expected and certain features were planned accordingly, but weren't able to be developed in the time frame of the research.

Similar to the issues regarding not being used to the control schemes, some participants also experienced nausea while flying, or got lost easily whether they were flying or walking in the environment. Nausea related to motion sickness is much more common in using VR headsets, which should be considered when developing such an environment.

The issues about getting lost were expected, and certain features were planned to be implemented to prevent such problems.

However an unexpected issue was the comments about the environment being too distracting for some users. It was understood that visual attractions such as rotating skyboxes and customizable avatars can cause a loss of focus for certain users and this issue should be considered when developing a serious virtual environment.

6.2. Next Steps

During the research, certain features were prioritized above others and were found necessary to address certain issues that were faced. Although these weren't able to be developed in the time frame of the research, it was found necessary to address them for future work.

Below are the most requested features, and the features that were found necessary to address the issues faced, which require immediate attention.

- **Spectator/Observer Modes**

As per request and the observations regarding issues in adapting to the utilized control scheme, a spectator mode where the users can choose to view from other users' cameras was found to be necessary. On another form of spectating, a mode in which the user is guided through the environment not only by spectating but perhaps pre-placed cameras is also found to be necessary for situations such as juries where the jury members might be meeting with such an environment for the first time.

- **Map/Minimap**

Many participants were lost during the sessions, including the author/developer, thus it was found absolutely necessary to develop at least a minimap for the environment.

- **Proximity/Spatial Voice Communications**

Issues regarding being unable to hold multiple conversations in the environment were not addressed as a part of the research, but were found to be an important opportunity in using virtual environments.

- **Permanent Sessions**

Due to the distance education period being over by the time the multi-user prototype was developed, and technical problems regarding 24/7 dedicated hosting from the campus, a permanent virtual studio environment was not able to be studied. A social, digital studio space was aimed to address certain findings from the preliminary studies, but as the test environment was a senior-year architectural design studio where students were already busy with their work, this scenario wasn't able to be tested as it was not truly necessary.

- **Shaders**

Addressing the issues about model quality and certain requested view modes such as arctic and wireframe, shaders should be developed as soon as possible. A cross-section shader was also planned, but wasn't able to be developed in the time frame of the research, which could help produce views such as plans, elevations and sections inside the environment.

6.3. Regarding OVS

In parallel to the research, OVS was developed from scratch through the explained methodology. Issues were identified, translated into requirements, which were ultimately interpreted into features and integrated into the developed software. Feedback and development loops were held with the help of surveys, interviews and testing in a real studio environment.

The development was done individually by the author in parallel to the studies. Due to limited experience in game engines and programming, and being within a limited time frame, not all of the requested and planned features could be developed in this period. Certain prioritizations were made, and a basic multi-user prototype was developed and tested.

OVS will continue to be developed, with plans for commercialization. Further research on the use of *multi-user environments in architecture,* concerning various scenarios and remaining features in the list, is also planned to be conducted by the author.

On another aspect of such environments, which wasn't discussed thoroughly in the scope of this research due to various reasons, a similar research focused on the use of VR in architecture was conducted in parallel to this research. This research was held by the thesis supervisor Tuğrul Yazar, with assistance from the author, and it was funded by the Scientific Research Projects Division of Istanbul Bilgi University. Such capabilities are being worked on, and are planned to be added to OVS in the near future.

As for the architectural design studios in Istanbul Bilgi University, OVS will be made privately available until commercialization, especially for the thesis supervisor Tuğrul Yazar's studio per their request, and the contributing tutors' studios as a token of appreciation.

6.4. Discussion

Regarding the third hypothesis: ***"****Architects have lagged behind in developing software for themselves. There are intellectual, productive and design barriers that arise from architects being limited to the tools that they did not produce or develop. In the case of producing tools and mastering the tools that produce tools, it is possible to perceive possibilities that cannot be grasped otherwise."*, self-observations were made during the period of the research.

One of the reasons for developing an environment from scratch was to see if doing so would help produce results that haven't been produced yet. During the state-of-the-art research, it became apparent that through having knowledge in programming and game development, and a development project in mind, one starts to play video games and use software in a certain way. Features that are found interesting or useful start to make you think of how they were produced, which can lead to reverse-engineering through testing and analyzing. On the other hand, certain features that are not related to the project at hand in any way, can bring up ideas for a new feature.

Size manipulation was one of them, playing a game completely unrelated to architecture, Psychonauts 2 (URL-27), and seeing the player character become smaller and go into the brain/mind of another character, the idea of scaling avatars instead of the imported models came into question. This was later found advantageous in terms of having users scale themselves instead of scaling a shared model, which would cause conflict otherwise.

Additionally, through getting more acquainted with programming and game engines, learning about certain functions and possibilities and following recent developments related to them can spark ideas of implementations for one’s field of work.

As an example, seeing that certain GIS software or services such as OpenStreetMaps or Google Earth’s APIs are available, in combination with some knowledge in game engines, one can immediately produce ways of importing such data into a 3D environment to be used for numerous reasons. Which is the case in Wooorld (URL-28), a multi-user VR software for exploring the world through satellite data, quite possibly from Google Earth. However there is a certain danger in such a way of thinking.

It was observed that as one gets more acquainted with programming, certain technical appeal and interests overshadow the seeking of reason and functionality. Interesting methods and features, regardless of necessity, can become a priority unless supervised.

On the positive side however, knowledge on such subjects in combination to knowledge in a certain separate field can lead one into finding ways to utilize their knowledge in one field, on the other.

REFERENCES

Aish, R. (2005). From Intuition to Precision. *Digital Design: The Quest for New Paradigms*, 5.

Bartosh, A., & Philip, A. (2019). Experimental Applications of Virtual Reality in Design Education. *Ubiquity and Autonomy*. ACADIA 38, USA.

Black, C., & Forwood, E. (2017). Visualisation and Analysis of Building Facade Movements as a Consequence of Loads on the Primary Structure. *Disciplines & Disruption*. ACADIA 36, Cambridge, MA, USA.

Braida, F., Chagas, I., Ruback Cascardo de Almeida, I., & Mendes de Castro, J. (2018). The Maker Culture and the Open Source Model in the Architecture, Urbanism and Design Context: The Fabrication and Sharing of a Game for Design Teaching. *Blucher Design Proceedings*, 1298–1304. https://doi.org/10.5151/sigradi2018-1629

Carpo, M. (2017). *The Second Digital Turn: Design Beyond Intelligence*. The MIT Press.

Centers for Medicare & MEDICAID Services (CMS). (2008). *Selecting a Development Approach*.

Chen, Z.-R. (2020). THE GUIDANCE SYSTEM OF GAMIFICATION AND AUGMENTED REALITY IN A MUSEUM SPACE. *RE: Anthropocene, Design in the Age of Humans*, 10.

Chien, S.-F., Lin, C.-I., & Lin, K.-L. (2020). A LIGHT-WEIGHT CAAD-VR BRIDGE. *RE: Anthropocene, Design in the Age of Humans*. CAADRIA 25, Bangkok, Thailand.

Du, J., Shi, Y., Mei, C., Quarles, J., & Yan, W. (2016). Communication by Interaction: A Multiplayer VR Environment for Building Walkthroughs. *Construction Research Congress 2016*, 2281–2290. https://doi.org/10.1061/9780784479827.227

Grasser, A., Parger, A., & Hirschberg, U. (2020). Pervasive Collaboration and Tangible Complexity in Realtime Architecture. *Anthropologic – Architecture and Fabrication in the Cognitive Age*, *1*.

Gül, L. F. (2019). Dijital Teknolojilerin İşbirlikli Tasarıma Olan Etkisi. *MEGARON / Yıldız Technical University, Faculty of Architecture E-Journal*. https://doi.org/10.14744/megaron.2019.23281

Gül, L. F. (2020). GELECEK, TEKNOLOJİ VE MİMARLIK. *dosya*, *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

Kandela, P. (2001). The buildings of Frank Gehry. *The Lancet*, *358*(9282), 677–678. https://doi.org/10.1016/S0140-6736(01)05769-5

LEITÃO, A., CASTELO-BRANCO, R., & SANTOS, G. (2019). Game of Renders: The Use of Game Engines for Architectural Visualization. *Intelligent & Informed*. CAADRIA 24, Wellington, New Zealand.

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. *Protocols, Flows and Glitches*. CAADRIA 22, Suzhou, China.

Nandavar, A., Petzold, F., Nassif, J., & Schubert, G. (2018). INTERACTIVE VIRTUAL REALITY TOOL FOR BIM BASED ON IFC. *Learning, Prototyping and Adapting*. CAADRIA 23, Beijing, China.

Oxman, R. (2006). Theory and design in the first digital age. *Design Studies*, *27*(3), 229–265. https://doi.org/10.1016/j.destud.2005.11.002

Pienaru, M.-I. (2018). The City as a Playground—Game tools for interactive planning. *Computing for a Better Tomorrow*. eCAADe 36, Lodz, Poland.

Rocha, A. J. M. (2004). *Architecture Theory, 1960-1980: Emergence of a Computational Perspective* [Ph.D. Dissertation]. Massachusetts Institute of Technology, Dept. of Architecture.

Sandstrom, A., & Park, H.-J. (2019). REFLECTION IN ACTION. *Intelligent & Informed*. CAADRIA 24, Wellington, New Zealand.

Santacruz, A. B. (2019). The Architecture of ScarCity Game. *ARCHITECTURE IN THE AGE OF THE 4TH INDUSTRIAL REVOLUTION*, 8.

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*.

Valls, F., Redondo, E., Fonseca, D., Garcia-Almirall, P., & Subirós, J. (2016). Videogame Technology in Architecture Education. In M. Kurosu (Ed.), *Human-Computer Interaction. Novel User Experiences* (Vol. 9733, pp. 436–447). Springer International Publishing. 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. *ARCHITECTURE IN*

THE AGE OF THE 4TH INDUSTRIAL REVOLUTION. eCAADe 25, Wiesbaden, Germany.

Yazar, T. (2009). *Mimari Tasarım Stüdyolarında Sayisal Egzersizler* [Ph.D. Dissertation]. Yıldız Technical University.

URL REFERENCES

URL-1. KooZA/rch.
https://www.koozarch.com/
[Accessed on 06.10.22].

URL-2. Google Drive.
https://drive.google.com/
[Accessed on 06.10.22].

URL-3. Google Docs.
https://docs.google.com/
[Accessed on 06.10.22].

URL-4. Lumion.
https://lumion.com/
[Accessed on 06.10.22].

URL-5. Spatial.
https://spatial.io/
[Accessed on 06.10.22].

URL-6. Unity Solutions & Services.
https://unity.com/solutions
[Accessed on 06.10.22].

URL-7. Autodesk Acquires The Wild.
https://adsknews.autodesk.com/news/autodesk-acquires-the-wild
[Accessed on 06.10.22].

URL-8. Agile Manifesto.
https://agilemanifesto.org/
[Accessed on 06.10.22].

URL-9. Minecraft Education Edition.
https://education.minecraft.net/en-us
[Accessed on 06.10.22].

URL-10. Cyberpunk 2077.
https://www.cyberpunk.net/
[Accessed on 06.10.22].

URL-11. Deep Rock Galactic.
https://www.deeprockgalactic.com/
[Accessed on 06.10.22].

URL-12. Grand Theft Auto: Online.
https://www.rockstargames.com/gta-online
[Accessed on 06.10.22].

URL-13. Grand Theft Auto V.
https://www.rockstargames.com/gta-v
[Accessed on 06.10.22].

URL-14. Davigo.
https://www.davigogame.com/
[Accessed on 06.10.22].

URL-15. Davigo Screenshot.
https://store.steampowered.com/app/1116540/DAVIGO/
[Accessed on 06.10.22].

URL-16. Phasmophobia.
https://kineticgames.co.uk/
[Accessed on 06.10.22].

URL-17. Valheim.
https://www.valheimgame.com/
[Accessed on 06.10.22].

URL-18. Risk of Rain 2.
https://www.riskofrain.com/
[Accessed on 06.10.22].

URL-19. Arkio.
https://www.arkio.is/
[Accessed on 06.10.22].

URL-20. Second Life.
https://secondlife.com/
[Accessed on 06.10.22].

URL-21. OpenStreetMap.
https://www.openstreetmap.org/
[Accessed on 06.10.22].

URL-22. Cities: Skylines.
https://www.paradoxinteractive.com/games/cities-skylines
[Accessed on 06.10.22].

URL-23. Unity3D Cross Section Shader Using Shader Graph.
https://codeburst.io/unity-cross-section-shader-using-shader-graph-31c3fed0fa4f
[Accessed on 06.10.22].

URL-24. Monster Hunter: World.
http://www.monsterhunterworld.com/
[Accessed on 06.10.22].

URL-25. Git SCM.
https://git-scm.com/
[Accessed on 06.10.22].

URL-26. Unity Netcode for GameObjects.
https://docs-multiplayer.unity3d.com/netcode/current/about
[Accessed on 06.10.22].

URL-27. Psychonauts 2.
https://www.doublefine.com/games/psychonauts-2
[Accessed on 06.10.22].

URL-28. Wooorld.
https://www.wooorld.io/
[Accessed on 06.10.22].