COLLABORATIVE VISUALIZATION AND VIRTUAL REALITY IN CONSTRUCTION PROJECTS

by

Juan Gonzalo Cárcamo Zuluaga

A thesis submitted to the

School of Engineering

in conformity with the requirements for the degree of

Master of Engineering with focus on Educational Technologies

EAFIT University

Medellín, Antioquia, Colombia

February 2014

Contents

Abstract

In the Colombian construction industry it is recognized as a general practice that different designers deliver 2D drawings to the project construction team. Some 3D modeling applications are used but only with commercial intentions, thus wasting visualization tools that facilitate the understanding of the project, that allow the coordination of plans between different specialists, and that can prevent errors with high impact on costs in the construction phase of the project.

As a continuation of the project "immersive virtual reality for construction" developed by EAFIT University, the present work intends to demonstrate how a collaborative virtual environment can be helpful in order to improve visualization of construction projects and achieve the interaction of different specialties, evaluating the impact of collaborative work in the design process of the same.

The end result of this research is an application created using freely available tools and a use case scenario on how this application can be used to perform review meetings by different specialist in real time. Initial test on the system has been made with civil engineering students showing that this virtual reality tool ease the burden of performing reviews where traditionally plans and sharing the same geographical space were needed.

Chapter 1

Introduction

The construction industry in Colombia generates a high impact on the economy, 9% contribution to GDP is estimated to related sectors and occupation of about 6% of the employed population [12]. However construction is characterized by a degree of underperform compared to other industries in variables such as product quality, industrial safety, environmental impact and meeting deadlines. Research projects in the field have shown the influence of the flaws in the design process and project planning in the performance of the industry, a situation generated in the growing complexity of projects, reduced execution times, and informality of the production system.

The construction industry in Colombia is known by the slow appropriation of technological developments and ICT commonly used in other industrial sectors, wasting a great opportunity to improve its performance. Building Information Modeling (BIM) and virtual reality models can contribute to the understanding of construction projects, facilitating its planning and execution, improving the final project performance, by detecting inconsistencies in pre-construction phases of the construction sequence, detecting interferences between different specialists designs and improving visualization, allowing a right implementation of the project.

As a result of the project being advanced by research groups in virtual reality and construction management, in late 2010 the infrastructure for virtual modeling and visualization in virtual environments on a large format projection screen was available. This work intends to show how using currently available infrastructure, 3D modeling capabilities and collaborative work between specialties of civil engineering, architecture and / or different research groups or companies can improve the design processes.

1.1 Contributions

This work presents the following contributions to the Architecture Engineering Construction sector in terms of:

- Showing the advantages of a 3D virtual environment for the design review process in terms of better understanding of the project, error detection between specialties and validation of a construction project..
- Details the design and implementation of an application for BIM 3D model's review.

1.2 Outline

This document is organized as follows:

Chapter 2 provides a review of the state of art related to collaborative virtual environments in the Architecture Engineering Construction sector.

Chapter 3 explains in detail the development of the Virtual Reality Collaborative Environment for reviewing BIM models, starting with the requirements of the development, initial architecture, implementation details and final product.

Chapter 4 describes a set of experiments carried out to validate the tool. The results of these experiments are detailed and the findings are interpreted.

Chapter 5 shows the conclusions of this work, some future related work involving the construction industry.

Chapter 2

Background

2.1 Immersive Virtual Reality

Immersive Virtual Reality has its beginnings in the mid-eighteenth century when the British scientist Charles Wheatstone developed the stereoscope, a device that allowed sight of a three-dimensional image, giving the illusion of depth. by superimposing two images of the same site taken from two different angles. Later in the year 1929 the first mechanical flight simulator was developed, "The Link Flight Trainer" which was used from 1934 as a mean of training for the newly developed instrument flight in military aviation, allowing realistic flight conditions without endangering the lives of pilots requiring training. This was the first mechanical device with ability to simulate real processes [13].

In late 1950s the first machine to create virtual spaces based on three-dimensional images called "Sensorama Simulator" was created, developed by Morton L. Heilig, this machine generated an illusion through sensory experiences from the movement of the observer's chair, from the the ability to generate wind in the observer?s face, and from a surround sound misleading the senses of the observer making him part of the projected image. In the 1960s instruments such as Head Mounted Display (HMD) or personal projection helmets were developed [13].

By the late seventies and early eighties, SUBLOGIC company developed the first flight simulator based on Mac personal computer systems from Apple Inc,being the first element of mass spreading virtual reality. Similarly planetariums were also some of the first to users of virtual reality systems, projecting cosmos' images in large sphered areas simulating sky's vision in a cloud-free night.

The term "virtual reality" can be traced to a literary work of 1938 ("The Theatre and Its Double"), in which Antonin Artaud described theater as a virtual reality. However, in the context of a three-dimensional, interactive computer generated environment in real time, the term was widely reported in the late 80's by Jaron Lanier. Jaron Lanier founded VPL Research Company in 1985, the company built the first virtual reality gloves and goggles[14].

In the early nineties, Carolina Cruz-Neira developed the first CAVE system® at the University of Illinois at Chicago in which images were projected across three screens in order to get a sense of reality for the viewer.[13].

A virtual reality user in a 3D world, the set between emotions and feelings that the person experiences as surround sound, off screens display of objects, among other experiences is what allows complete immersion in the 3D virtual reality world.

One of the main benefits that represent the virtual reality is that allows the user to come face to face with the information, e.g researchers can explore, the interior of a DNA molecule with a simple joystick or virtual glove, and if wanted a pair of stereoscopic glasses and a stereoscopic display, allowing them to be as close as desired to their particular point of interest.

2.2 Technical considerations of virtual reality systems

Typically Virtual Reality applications require the following items for its construction[15]:

- 3D scene loaders: Computer programs to read the descriptions of the 3D scene. A 3D scene is composed of one or more files which contain descriptions of geometries, textures, images, sounds, and position in the virtual environment, also known as the 3D scene objects.

- Navigation Mechanisms: Software programs that allow the user to move between objects in the 3D scene. Some of these forms of navigation are: Navigation with 6 degrees of freedom (up, down, forward, backward, left and right), browse a predefined path, move an object in the 3D scene without modifying the user moving their rotation, among others.

- Collision handling: If navigation mechanism need it there should be a computer programs that handle user restrictions against objects within the virtual environment such as object collision with the floor and walls if the object is an architectural tour, or perform some action such as opening a door once the user is in front of it to access another 3D scene.

- Objects Animation: 3D scenes can contain dynamic elements, either by the virtual reality program performing the movement of some object within the 3D virtual world, or recorded animations created by software such as 3D Studio Max or Autodesk Maya. The virtual reality environment should be able to play these animations.
- Physics simulation: Side by side with the collisions handling item, and to add realism in the virtual reality environment, physical simulations may be needed such as gravity, water, wind.
- Characters interface: A special case of 3D objects are the characters. These may be the representation of the user in the virtual world and interact within the same, or may be part of the environment and add realism to it. When needed to be part of the world it is necessary to include artificial intelligence programs that simulate complex behaviors, such as a virtual guide in the world. It is important to note that the characters do not necessarily have to be anthropomorphic, they can be shaped as animals, robots or other non existing being that plays the role of character within the world.
- Spatial sound: A mechanism for increasing immersion in a virtual world is the inclusion of ambient sounds in it, for example the steps of the characters, auditory feedback when they are at a point of interest and other.
- Human computer interaction interface: In addition to the keyboard and mouse, virtual reality programs can make use of various hardware devices to interact in 3D scenes, for example, controls from video game consoles, optical tracking systems to determine the position within the world, electronic gloves for 3D object manipulation, force feedback devices (called haptics), among others.

2.3 Building Information Modelling (BIM) [1]

Building information Modelling can be described as the mix of technologies and a methodology to improve the architecture engineer and construction (AEC) product development by reducing project delivery time, increasing productivity and quality, and decreasing project cost.

At its heart BIM methodology is based on Building Information Models. These are digitally constructed models that contains precise geometry and relevant data that supports the various activities in the AEC industry, such as design and construction.

Commonly BIM models contain:

- Geometry information
- Spatial relationships
- Quantities and properties of building elements
- Cost estimates
- Material inventories
- Project schedule

The final goal of BIM is to encompass all aspects of AEC process within one virtual model, allowing all the project participants to have a single, unified view of the project. This improves the collaboration among the different specialties that interact in this kind of projects.

2.4 Related Work

Several theoretical work has been done around Computer Assisted Design since the 80's. The following section presents an overview of some of the research carried out by the Architecture Engineering Construction sector regarding the construction project planning and design.

2.4.1 AEC Collaborative Information Systems: From Requirements to Architecture [2]

In this article the researcher present a framework on how a collaborative system should be made to support the complex dynamics in the AEC industry. Computers have supported various task the life-cycle of a construction project such as architectural modelling, structural design and project scheduling. In order to create an effective collaborative environment the author proposes the following capabilities of the system:

- Provide design and management tools.
- Coordinate participants over the life-cycle of the project.
- Serve as a knowledge integration tool.
- Avoid decision conflicts that can lead to failures.
- Create a change history for future use an liability tracking.
- Provide a basis for evaluating proposed changes.

2.4.2 Information Model for Managing Design Changes in a Collaborative Environment [3]

In this paper the authors discuss several techniques used to perform building designs outlining the process commonly involved in managing building design changes. These techniques involve:

- Regular coordiantion meetings, where the different disciplines review other's designs to find information changes that can impact their work.
- Informing changes to all design team, once a designer makes a change it should inform all the other team members so they are aware of this changes, but this usually becomes impractical in large projects.
- Usage of checklist to verify compatibility, which usually prompts problems when one change affects several other designs. Besides keeping this checklist up to date therefore many incompatibilities are skipped using this technique.
- Usage of Computer Aided Design (CAD) software to generate technical documents. This technique is effective to avoid inconsistencies among various drawings (or models) but most of this software is only compatible with a single design discipline, limiting its value.

It is worth to note here that a lot of advances had taken place in the last decade, e.g Autodesk Revit now integrates Architectural, Structural and Electric designs in a single project. But still the mechanisms of keeping the project up to date among all participants still part of the problem.

After outlining this techniques the authors propose an information model to handle this design changes. This proposed model has the following characteristics:

- Help propagation of design changes to the affected disciplines.
- Track past design changes.
- Provide the ability to plan and schedule future design changes.
- Data over drawings constitute the main media for storing and communicate information.
- Single source of design information.

2.4.3 A/E/C Teamwork: A Collaborative Design and Learning Space [4]

This paper describes an Internet-based Web-mediated collaboration tool kit and a people based learning environment. Here the author describes how using internet technologies several disciplines geographically dispersed can participate in project and produce better, faster, and more economical products. The evaluation of this project was conducted with the AEC course offered at Stanford university.

The main characteristics of this communication toolkit are:

- Team building over the internet, where students choose their building project based on their skills, preferences and availability of the project.
- Cyberarchive for building projects case studies that can be accessed by anyone integrating a team project, faculty and students. This archive serves the purpose of orienting the different projects on particular issues concerning the disciplines of interest and at a different levels of detail. This archive consisted on images and AutoCAD files to be shared over the internet.
- Shared project workspace for each project team. This workspace allows the participants to archive, share, access and retrieve the different information of the project. This information

ranged from sketches, document files, AutoCAD drawings to VRML product models. This workspace also notified the team members about CAD related modifications.

- Team discussion forums to ease the capture, sharing and reuse of ideas within the project.

2.4.4 CADOM: A Component Agent-based Design-Oriented Model for Collaborative Design [5]

The authors start their research from the model representation discussion in building design, summarised in model representation concerned of the model content and representation means for supporting model production and sharing, and model management concerned with the data management during the design process. In the first scenario the most common representations are structure oriented models and design oriented models, being the latter the approach used by the researchers.

In design oriented models (DOM) information about the design purpose, function and reasoning is introduced in the model. This information is as important for the designers as it is structural data. To build this DOM a component agent mechanism is proposed.

Component agent DOM supports collaborative views by creating multiple-views for different disciplines while managing the different constraints within the design and managing the design life-cycle by handling the different versions and authors of designs.

2.4.5 Collaborative design of structures using intelligent agents [6]

Building design in the construction industry has a long tradition of collaborative work, however at the design stage this work is usually based on physical meetings between representatives of the different design disciplines, architecture, engineering, construction. In this paper the authors discuss how intelligent agents, known as self-contained knowledge-based systems, can improve collaboration among specialist of the same discipline with special focus on synchronous and asynchronous distributed collaboration.

For this end the authors used Zeus as an interoperability framework, allowing the creation of agents in a layer structured way. The components of this agents are the following: knowledge relating to a particular area of expertise and negotiation strategy for design changes ensuring overall agreement with the other agents.

With this approach the researchers believe that artificial intelligence offers major scope for facilitating collaborative working by addressing the integration of multidisciplinary perspectives while providing a framework for resolving design conflicts between a construction project team.

2.4.6 Multiple Device Collaborative and Real Time Analysis System for Project Management in Civil Engineering [7]

In this article the authors propose a Collaborative Distributed Project Management system focused on portability of the clients and specific to construction industry. This system was modelled using model-view-controller architecture and uses CORBA as it's network technology to communicate the different clients with the operations needed on the server side.

A real scenario of the proposed system is presented, showing how to dynamically analyse the project and collaboratively connect the different companies involved in this task, and also showing how structured meetings can help to maintain a single source of information, such as CAD designs, and in real-time visualise and decide over certain aspects of the design.

It is worth to note that this system's purpose is project management and offers no solution to technical and collaborative conflicts.

2.4.7 Building a project ontology with extreme collaboration and virtual design and construction [8]

In this work, ontology based project creation based on Product, Organization an Process (POP) generic ontology is presented. Aimed for Virtual Design and Construction (VDC) modelling and extreme collaboration (XC) methodology, the author intends to show how different disciplines can collaborate in building design improving project meetings by focusing on explaining and evaluating the designs, and making decisions instead of being focused on describing a design.

Although this paper doesn't present a solution on geographically distributed specialist, this approach is interesting in terms of sharing different types of information among specialist making use of extreme collaboration technologies and ontologies to describe the project components.

2.4.8 Implementation of IFC-based web server for collaborative building design between architects and structural engineers [9]

Industry Foundation Classes (IFCs) are a ser of building produce model specifications developed by the International Alliance for Interoperability (IAI) for product data representation and exchange in AEC industry. In order to have collaborative design between different specialties, the authors propose an implementation of a web server based on IFCs, taking advantage of the Object Oriented CAD design centred on geometry information from architecture and structural design.

The end result was a Java application able to handle restrictions on design following the IFCs constraints, enabling feedback from structural designers to architecture designers until they agree on a common design. Using Java 3D both specialties are able to view a 3D representation of the design and collaborate through the web server having always the same information on each end.

2.4.9 Using a shared workspace and wireless laptops to improve collaborative project learning in an engineering design class [10]

Using Basic Support for Cooperative Work (BSCW) and IBM laptops a class on engineering design of civil engineer department was studied by the authors in order to asses the use of shared resources between project groups. BSCW is a web collaboration system with features such as forums, file sharing, access privileges, authentication, search builder, versions management, among others.

This paper is relevant to the present work in the sense of knowing that there are already systems able to handle a lot of the commodity functionalities for cooperative work, as an indicator of the need to include this kind of technologies in the AEC industry processes.

2.4.10 State-of-the-art technologies and methodologies for collaborative product development systems [11]

With the advancements in Information Technologies (IT) collaborative work through all the product life-cycle is a must do. The authors of this paper summarise this kind of developments in three main categories:

- Visualization-based collaborative systems: Used to facilitate conceptual design or product preview or reviews. Its main purpose is to view parts of the model. Common functionalities for this systems are: viewing, markup, measure, among others.

- Co-design systems: Aiming to achieve complex design task synchronously or asynchronously, its main functionalities range from concurrent editing CAD files to perform detail design and review.

- Concurrent Engieneer (CE)-based systems: This systems purpose are to establish a communication channel between design and manufacturing, effectively handling the product life-cycle. Creating design team organisation, design model synchronisation, design conflict resolution are some of its common functionalities.

Several technologies and architectures have been proposed for all of this systems, worth noting Java Applets, Java 3D, Microsoft's ActiveX, HTML and XML were the most common technologies used, at the time of this article, to implement this systems on the client side part. On the server side programming J2EE and Microsoft's .NET are actively used. For the third kind of systems CORBA and Java/RMI help with the implementation of the communication that all the involved applications need to fulfill its purpose.

As standards for visualisation at the time of this article VRML, X3D, W3D and OpenHSF were the most commonly used for design web sharing.

To date February 2014, alternatives like Jmonkey have replaced the Java applet and Java 3D technologies.

Chapter 3

Contribution

In this chapter a detailed explanation on how the proposed system was built is given. First it starts with the definition of the goals of the project, then the main requirements of the desired visualisation-based collaborative system are described. With the goals and the system requirements in mind the reader is given an analysis of the tools considered to achieve these objectives. In the fourth section some technical considerations for the chosen tool are given. The fifth section describes a high level design of the application built. Finally it concludes with the implementation specifics of the developed tool.

3.1 Goals

3.1.1 General Goal

Create a collaborative work environment for real-time visualization using virtual reality generated from 3D BIM models, in order to determine the benefits and difficulties of implementing this new scheme of work in the development of construction projects.

3.1.2 Specific Goals

- Develop virtual reality 3D BIM models for visualization.
- Select student groups or construction companies to conduct experiments of collaborative visualization using the built 3D BIM models and the virtual reality infrastructure at EAFIT university and the company or research center participating in the project.
- Perform work in a collaborative real-time visualization for better understanding of evaluated construction projects, interference detection between specialties and validation of the tool.
- Assess the benefits and barriers of collaborative work by reviewing virtual reality models for the construction industry.

3.2 Main requirements

After "immersive virtual reality for construction" project was finished in 2011 one of the proposed future work was to adapt the technology available at EAFIT's virtual reality for construction lab to enable collaborative visualisation in real time between the different stakeholders of a construction project. This collaborative virtual tool should enable real-time visualisation of the BIM models produced by the lab for its review in different geographical locations. (synchronous communication over a computer network).

This tool should make use of the stereoscopic view settings of the lab and also provide a mechanism to point at an specific part of the design just as in a same geographical location review meeting with a laser point.

From the former definition a list of requirements separated in functional and non-functional requirements was created as follows:

3.2.1 Functional Requirements

The following functional requirements were identified:

- The system must allow a user to be a host of a review meeting.

- The system must allow a user to connect to an ongoing review meeting.
- The system should allow multiple users connect to the same review meeting and interact with each other.
- The system must allow the projection of stereoscopic images of the models.
- The system must use an optical tracking device that allows users to point out the different virtual objects that make up the model.
- The system must allow a modeler to upload his model created in Revit or 3ds Max for its review.

3.2.2 Non-functional requirements

It is expected that this collaborative virtual environment meets the following non-functional requirements:

- Security: The system should provide reliable information, the information must maintain its integrity and must be available when a user needs it.
- Scalability: The system must be constructed so that in the future more functionality may be added, without affecting system performance.
- Robustness: The system should be able to support exponential growth of users.

3.3 Tools alternatives analysis

As seen in chapter 2 a lot of advances have been developed in the Computer Graphics field. While doing this project the author needed to test several alternatives for the implementation of the needed application. From low level programming languages and specifications to high level, design oriented software development platforms, different analysis were conducted. In this section is presented a brief summary of the tested tools before beginning the development.

3.3.1 DirectX and OpenGL

Interacting with the computer peripherals is not always an easy task. Usually hardware vendors provide software applications called drivers for this purpose. In the case of Graphical Processing Units (GPU) the drivers are made compliant with OpenGL and DirectX Application Programming Interfaces (API). DirectX is a set of APIs provided by Microsoft to create high-performance multimedia applications [16]. OpenGL is a hardware independent specification of a system that provides an API that serves as a computer graphics rendering system itself, meaning it could be implemented entirely in software but designed to be hardwired implemented[17].

For the purpose of this work this two APIs were compared against the requirements. In the first place this two APIs only deal with graphics rendering, meaning that every other functionality needs to be done through other APIs. Learning these technologies could be very time consuming and perhaps the end result might not be as reliable as expected.

In the case of OpenGL creating a window to display the 3D application is a burden and it's tied to the specific operating system where it is being implemented. Several libraries exist to ease this problem such as SFML, SDL, GLFW, freeglut, OpenGLUT among others but still is a major drawback for development in terms of the learning time it takes to get started. On the other hand DirectX APIs are designed to work only under Microsoft's Windows Operating System, this gives certain degree of interoperability with other APIs such as the Windows Networking API, but it still time consuming in development complexity.

The advantage in using this APIs is having total control over the development where any unexpected behavior can be controlled, however the cost of producing this tools is out of the scope of the present project as there are other higher level development tools that could help in the construction of the intended system as explained further in this chapter.

3.3.2 EON Studio

EON reality is a company focused on the development of 3D author applications. Their main product is called EON Studio. With this development software users not familiarised with 3D application development can create high quality interactive 3D content [18]. It can import a lot of the commonly used sharing 3D content formats, such as 3ds, obj, dwf 3D, VRML among others. It is also ready to make use of built-in tracking systems, networking and stereoscopic projection (subject to licensing).

EAFIT university currently owns a licence of EON Studio, with the peripherals (tracking system) and stereoscopic view licences. This tool has proven to be effective in the creation of 3D environments for presencial BIM models review. Along with Autodesk's Revit and 3ds Max it is part of the set of tools currently offered by EAFIT to the construction sector.

The advantages of using EON Studio from this work's point of view are: Easiness to import different models formats, built-in navigation system, "Drag and Drop" 3D programming through the use of visual nodes and scripting language to enhance functionality using JavaScript or VBscript syntax.

As previously stated, this development tool offers a lot of the functionality needed for a collaborative review tool, unfortunately it is a proprietary licenced software, meaning no enhancements could be done unless buying them from the manufacturer, like the networking component. It also lacked of proper documentation, making the development experience more of a trial and error kind of situation, very frustrating and time consuming.

3.3.3 Unreal Development Kit (UDK)

UDK is the free version of the award winning 3D game engine called Unreal Engine 3 from EPIC Games Inc[19] used in hundreds of AAA quality games. It has all the functionality of Unreal Engine without the full access to its source code[20]. Some of the most important features related to this work are: Networking capabilities, RealD stereoscopic view, Kismet visual scripting, and UnrealScript programming capability.

Architectural walk-throughs can be done with UDK, it has a wide development community and a lot documentation. It is also free to use for educational and noncommercial uses, and it has a very affordable licensing scheme for commercial purposes.

On the other side, its learning curve is really steeped. Development needs very experienced people in game production pipeline to produce an end product and the import of 3D models is really troublesome, needing to deal texturing with of each object within the engine. Also no documentation on how to integrate with other peripherals as tracking systems were found, this could be done through dlls according to its documentation but further studies need to be conducted.

3.3.4 Game engine - CryEngine

Developed by CryTek, CryEngine is a powerful AAA quality game engine with fully integrated physics, weather, particle effects, terrain editing, dynamic vehicles, modular buildings and roads, and complex animation systems[21]. Concerning this project, CryEngine has an specific platform called CineBox, with support of full stereoscopic pipeline, fully linear rendering pipeline and import and export different 3D scene formats including Autodesk's fbx.

Its major limitation is the lack of documentation to get started with the development, the tutorials only cover how to set up a scene, but as how to use its networking capabilities, or how to integrate with other peripherals its documentation falls short.

3.3.5 Game engine - Unity 3D

Unity 3D is a game development ecosystem which includes rendering engine and a complete set of tools designed for rapid workflows between designers and programmers to create interactive 3D and 2D content [22]. Unity has two versions: free edition and pro edition. The licensing for the pro edition depends on the functionality wanted, e.g pro plug-in for android, pro plug-in for iOS, version control. Its main attractive is its multiplatform support, from the same project a 3D content creator can publish its work on PC, android, iOS, Xbox 360(if it is a licensee), among others. Its free version is highly appreciated among the Indie developers' community, having the most active community to date.

Unity allows to develop all the requirements of this project in its free version with only one limitation on the stereoscopic view. This limitation concerns with the kind of stereoscopy technique that can be implemented. Regardless this limitation, the networking capabilities and the 3D content creation workflow make this platform the ideal alternative for this project.

3.3.6 Chosen tool: Unity 3D

After reviewing the alternatives the most fit platform to develop this tool was Unity 3D. Not only it had the most fitting workflow, but it's rendering features are built upon DirectX and OpenGL for platform compatibility, its programming model is easy to understand not only by programmers but also for designers, it has a very simple integration with dlls within the system that helps communication with peripherals such as OptiTrack tracking system and Microsoft's Kinect and its built-in networking model is adequate for the work at hand. The table 3.1 on page 24 summarise the former analysis.

3.4 Networking in Unity 3D

In computer science networking is defined as two or more computers sharing information between them. It has never been an easy task and several proposal has been made since the invention of computers[23]. In terms of virtual reality environments, networking concerns with sharing relevant data to maintain consistency in the perceived world. This includes, but is not limited to, position, and rotation of the characters in the scene, allowing all the connected people perceive the same changes in 3D Objects made by each participant and written or spoken communication among participants.

Unity 3D provides an easy to use interface to design and create network interactions, but even so certain considerations need to be taken into account when working on networked projects. Every choice made has an impact in bandwidth consumption, data processing, jitter, delay of updating the environment (commonly know as lag) [24]. The previous items when handled wrong can be perceived by the end user as broken software.

In virtual reality networked environments a client-server approximation is usually taken. A server is a computer software that handles information request from clients such as resources available, heavy computations needed and user's state information. The client is a computer software that communicates with a server and its work is driven by requesting information from the server to present it to the end user. In this section the high level concepts of Unity 3D networking are explained.

3.4.1 Networking approaches

In Unity 3D two approaches to achieve networking are used. Authoritative servers or a design where everything that happens in the network environment is handled by the server, and non-authoritative

Requirements	OpenGL Development Readiness	DirectX Development Readiness	EON Development Readiness	UDK Development Readiness	CryEngine Development Readiness	Unity 3D Development Readiness
Allow a user to host a review meeting.	No	No	Yes with network license	Yes using UnrealScript	Yes	Yes
Allow a user to connect to ongoing review meeting.	No	No	Yes using network license	Yes using UnrealScript	Yes	Yes
Allow multiple remote users connect to the same review meeting and interact with each other.	No	No	Yes with network license	Yes using UnrealSript	Yes	Yes
Allow the projection of stereoscopic images of the models.	Yes	Yes	Yes using stereo license	Yes using RealD	Yes	Yes with limitations
Use an optical tracking device that allows users to point out the different virtual objects that make up the model.	No	No	Yes with peripherals license	Yes	Yes	Yes
Allow a modeler to upload his model created in Revit or 3ds Max for its review.	Yes	Yes	Yes	No	Yes	Yes

Table 3.1: Readiness analysis summary

servers where clients are responsible for the changes in their environments and notify the server about this changes so it can notify them to other clients who are responsible for making this changes.

Authoritative servers are used in games to avoid players from cheating, as every movement within the game has to be accepted first by the server, malicious modifications of a game will not be able to perform illegal actions within the networked session. This approach is also used in physics simulations to avoid the users seeing different things on each client, since the server is the one performing the changes in the 3D scene.

The disadvantages of authoritative servers are the overhead from processing all data from all clients making the simulation slow for clients and the need to send every action to the server and then getting back the results produces a delay in the execution of the simulation which can be perceived by the user, something that can be annoying to the user and takes them out of the immersion. To solve the former bigger processing machines need to be used to run the server program. To solve the latter client side prediction algorithms are implemented so the user don't feel the network delay.

Non-authoritative servers on the other hand gives the clients full control of the simulation at hand thus allowing the user to manipulate the environment at his will. In this approach the server works as a message broker, sending messages to all clients about the simulation state. This implementation is easier than the former one. For this work's purpose a mix between authoritative and non-authoritative server was proposed.

3.4.2 Network communication

Two methods for network communication can be used within Unity 3D. The first one is Remote Procedure Calls (RPC) which are messages that invoke a method on from one computer to other over the network. The second one is State Synchronization which are messages sent from the client to the server and relayed to all the clients to keep information about a 3D object updated across the network. It is common to use both methods while implementing the networking functionality in Unity 3D.

RPC are used for infrequent actions such as opening a door, or in the case of this project point at an object. This actions are not constantly happening as much as, for example, the user's position within the environment. On the other hand, actions that are constantly taking place such as changes in user's position and orientation are effectively handled by State Synchronization, as the server is relaying this information to all the clients there's no need for the server to invoke an specific method on each client instead each client that get this message needs only to update the desired information in the local environment.

It is important to be careful with which 3D objects are subject to state synchronisation as it can take a lot of bandwidth from the network. Special attention to this part in the design needs to be done.

3.4.3 Some other aspects

Connecting a server and a client is complex task since there are network constrains such as private and public IP addresses, local or external firewalls, closed ports over the network among others. Unity provides tools to test different network situations such as client with IP private address, server with IP public address, both server and client with IP private address given they both have internet connectivity, and both client and server having public IP addresses. After establishing that connection is possible Unity 3D provides two methods to establish the connection: Direct connection where the client knows the IP address and port of the server, and Master Server connection, a special server where network virtual environment servers advertise themselves to clients.

Master Server is an application provided by Unity Technologies where servers developed by Unity 3D users can register themselves. After registration clients can ask the Master Server Utility the needed information to connect to an specific server. Not only this avoids the end user to know the specific details about networking connections in the virtual environments, but it also allows several teams to connect to different servers of the same application to perform separate actions. In this work, this functionality could allow different review teams to be reviewing the same BIM model on different servers without interrupting each other or causing network overhead on only one server.

3.5 Tracking systems in Unity 3D

To find the physical location of a person and to use it in a virtual environment, tracker devices are used. Its responsibility is to copy the rotation and translation of the user within the virtual world.

If tracking is done with a fixed coordinate system is called absolute tracking, on the other hand if tracking measures only the incremental movement is called relative tracking[25].

Optical Tracking, or tracking systems based on cameras, can be done with or without markers and this depends on the system's technology. Two technologies were tested by the author, Natural Point's OptiTrack solution and Microsoft's Kinect. Both of this solutions could be used within Unity 3D development environment.

3.5.1 Natural point's OptiTrack VRPN

OptiTrack solution is an absolute tracking with markers system. It is composed of NaturalPoint's v100 infrared cameras that can capture 100 frames per second with a latency of 10 milliseconds [26]. Along with the cameras Natural Point offers a software called Arena that can track the markers from the cameras and it can work as a 3 different servers that provide the position of the markers in the space.

The server used in this test for interfacing the tracking system with Unity 3D is a VRPN server. VRPN stands for Virtual Reality Peripheral Network, a collection of libraries to allow connection of different peripherals of a virtual reality environment in a client-server basis. This abstraction enables the manipulation of different devices in one host computer[27].

The first step was to calibrate the cameras. After this calibration the VRPN server of Natural Point's Arena software was streaming accurate information.

To interface the cameras with Unity a proprietary middleware called Middle VR was used[28]. This middleware creates an interface within Unity 3D and VRPN servers. Its setup is done by the Middle VR application, after connecting to a VRPN server a configuration file is created to be used within Unity 3D.

The tracking of the markers worked nicely with the provided examples, and the integration of this functionality with a different 3D scene was fairly easy to do. The only problem was that this solution only worked with Unity 3D pro version, plus the use of the middleware has a cost of 3000 euros, making it unusable for this project.

Other alternatives to connect with the VRPN servers were considered such as UIVA, Unity Indie VRPN Adapter, an open source implementation of a client, server for using VRPN within Unity 3D free edition [29] but its development time made other alternatives to be looked at.

3.5.2 Microsoft's Kinect

Microsoft's Kinect is a markerless tracking system. It consist of an array of two cameras, one RGB and one infrared, an infrared light emitter, an array of microphones and an accelerometer to determine the orientation of the device[30]. Originally created for the video game console Xbox 360, it has found many applications in computer programs by the existence of the software development kit (SDK) from their makers and a growing open source community called OpenNI. This system is able to track 2 users in 19 joints[31].

Three approaches were found to use Kinect with Unity 3D, using FAAST VRPN server [32] with UIVA[29], which is already supported by this solution, Using OpenNI driver and open source wrapper developed by Zigfu [33] and Carnegie Mellon University Entertaintment Technology Center Microsoft's SDK wrapper [34].

The first and the third approach only work under Microsoft Windows environment. The second one works under Windows and Mac OSX environments.

To test this solutions the provided examples by the developers were tested first, then an attempt to use the provided APIs under a new 3D scene was made. The three solutions worked with the examples. Some points worth noting here is that the UIVA and Zigfu approaches needed calibration from the user to start working, something that newer versions of the Microsoft's SDK already had solved. With UIVA some delay was experienced while performing the tracking of a body, this behaviour could be explained by the need of receiving data through a VRPN server.

Zigfu also offers a paid version of its kinect wrapper, but as mentioned previously on this document it was out of scope.

Installing the different components needed for the first and second approach was seen as a possible set back for the users to use the proposed solution, making the use of the third alternative the one used for the application development.

3.6 Stereoscopic view in Unity 3D

Regarding computer graphics, several techniques are used to produce stereoscopic images, commonly known as 3D due to its recent use in the film industry. From anaglyph projection of images to more sophisticated techniques such as frame sequential stereo rendering 3D content is being created to increase realism in virtual reality environments. This section describes some of the most common 3D features used in today's environments and describes how to implement one of them in Unity 3D.

3.6.1 Anaglyph stereoscopy

This technique takes two images, one for the left eye focus and another one of the right eye. Processing the images by filtering one's red color and the other one's blue or green color (or a combination of both which is cyan color). After filtering the images, both images are superimposed into one single image. The resulting image can be then view as an stereoscopic image using glasses with cheap filters for the colors that were removed from each picture, commonly red for the left eye and cyan for the right eye [35].

The disadvantage of this method comes from the weird colors the resulting images end with, and even though a lot of improvements have been made in this area it is not an optimal solution for the needed application.

3.6.2 Side by Side stereoscopy

In computer graphics, by using two projectors with polarising filters, an special screen dubbed screen silver which accurately reflects the polarised images projected on them, and special glasses with the same polarisation as the one applied by the screen [36], a viewer can have the sensation of being immersed in a virtual reality environment. To achieve this, the computer generates two images side by side or over and under in a single screen[37], then using special hardware to split the output of the screen in two outputs and pointing the projectors to the silver screen this effect is achieved. Most recent 3D ready hardware such as 3D television sets and 3D projectors can emulate the use of two projectors with filters.

The needed images have to be taken as if they were taken from the right and left eye position respectively. In this method no processing of the images needs to be done in the computer, as the filtering is being made by the filters at the projector and the glasses.

The projection of this images are at best half of the resolution of the projection system capabilities since the screen needs to be cut by half to host the left eye and right eye image. To solve this problem an advanced technique dubbed quad buffering stereo is used.

3.6.3 Frame Sequential stereoscopy - Quad Buffer stereo

Image quality can have a great impact in the immersion sensation a user has while being in a network virtual environment. In order to create high definition images are of 1920x1080 pixels or higher. With the side by side approach producing this quality images need a very high end computer graphic's card capable of rendering this two images simultaneously which is often not the case. This approach looks for the presentation of images sequentially in time, first for the left eye and then for the right eye, with the proper equipment this images can be projected synchronically in the correct order and fast enough for the user not to notice the change[38]. Currently the flow of this two images is made possible with a quad buffer feature of high end video cards hence the name quad buffer[39].

Usually this stereoscopy method can use two types of glasses active, as in using batteries for polarising the glasses, or passive where no batteries are needed. Active glasses are synchronised with the host computer or screen projecting the virtual environment to ensure synchronisation of the polarisation with the projected images[40]. Passive glasses on the other hand are less expensive but the resulting stereoscopic effect tends to be less powerful.

3.6.4 Implementing stereoscopy in Unity

As mentioned early, stereoscopy in Unity 3D had one disadvantage: not having support for quad buffered stereo. Although this presents limitations in the quality of images presented to the user, the goal of this project is to present a tool for collaborative review of designs. In the AEC industry other tools like 3Ds Max are used to render high quality images, and usually while reviewing a design is more important to notice interference between models than texture details or extremely realistic environments.

Paul Bourke offers a guide on how to implement side by side stereoscopy within Unity 3D[36].

Using 2 perspective cameras, two side by side planes and an orthographic camera side by side stereoscopy is achieved. The implementation is very straight forward but it needs Unity 3D Pro in order to work. Another tool was tested to achieve stereoscopy called FOV2GO[41]. This tool provides the needed functionality in Unity free to properly work with side by side stereoscopy.

3.7 High Level Design

To start with the development of the proposed system a high level design was made to serve as a roadmap for the desired system. This design was made using UML and a short description is given for each diagram component.

It is worth to note that the requirements regarding being a host of a review meeting are being treated as "server functionality" within the design, while the connecting to a review meeting requirement is treated as "client functionality".

3.7.1 Use Case Diagram

The use case diagram intends to present how the users will interact with the system. Figure 3.1 shows the design created for the collaborative review virtual environment based on the requirements. Two actors are identified in the system:

Participant: This user will be the one conducting the review model. Their responsibilities are to navigate the environment and find errors models loaded in it.

Modeller: Is responsible for the content to be revised in the collaborative virtual environment. In this role the user must insert the various models in the virtual environment and position them within the same.

3.7.2 Class Diagram

A class diagram is an static representation of the types of objects being modelled. This representation shows the associations between the different types, and in further refinements contain each classes' attributes and actions, to latter be programmed as a fields and methods according to the Object Oriented programming paradigm. Figure 3.2 shows the high level class design for the proposed

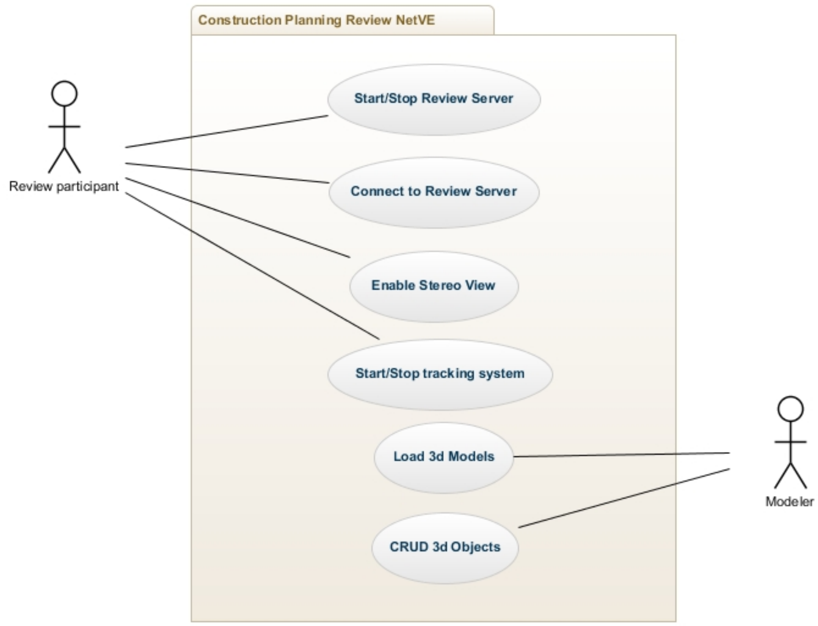


Figure 3.1: High Level Use Case Diagram

application.

In first place the package called Unity 3D represents the embedded functionality that comes with the IDE off the shelf. Outside the package, four classes are proposed to model the system: Review participant, network manager, tracking manager and stereo camera. It is the author's belief that this classes are sufficient to illustrate the functionality of the review virtual environment.

Briefly explained a review participant has an stereo camera to enable stereoscopy when needed, a network manager that will allow it to be server or client of the review meeting and will handle the communication between clients, and a tracking manager that will handle the information provided by the tracking system.

3.7.3 Sequence Diagram

At this point, the static elements of the system have been described. In this subsection a dynamic modelling diagram is used to illustrate the desired system's behaviour. The table 3.2 on page 33 shows the main behaviours in the system ordered by the use case they support.

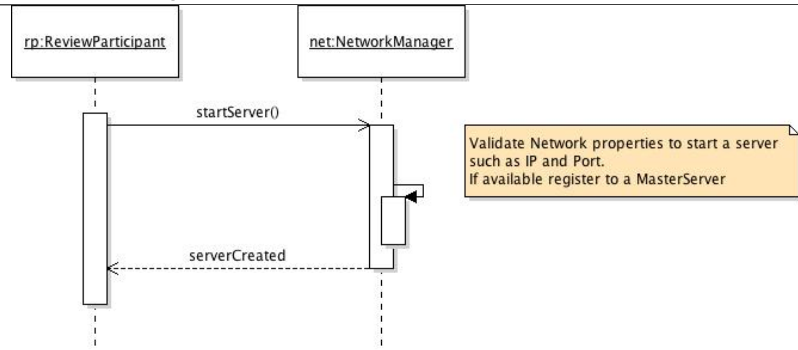
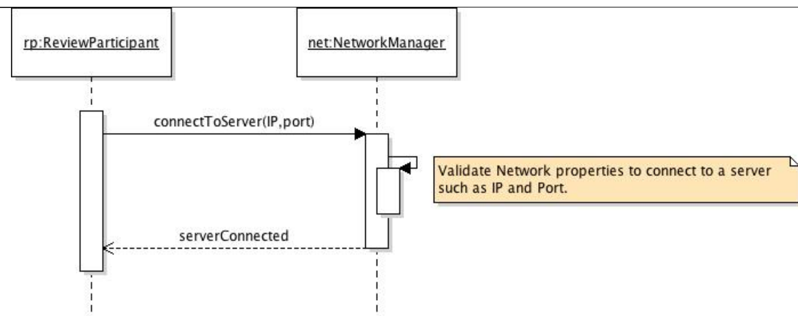
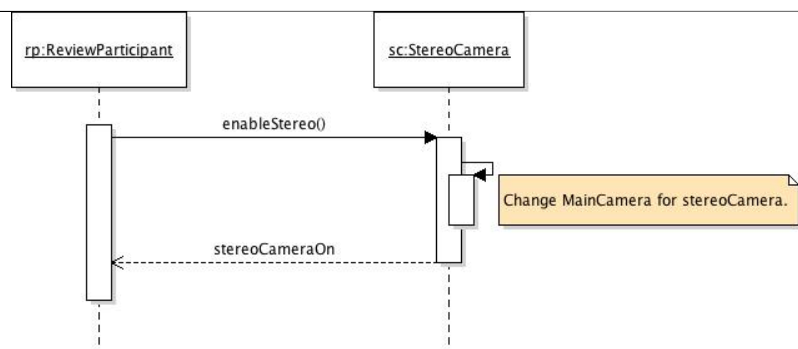
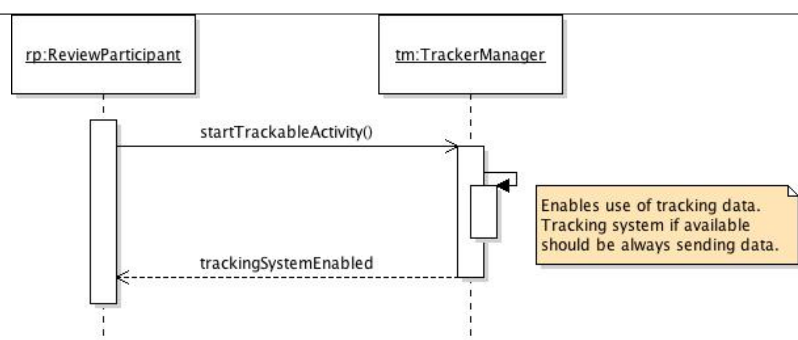
<table>
<tr><th>Use Case</th><th>Sequence Diagram</th></tr>
<tr><td>Start Review Server</td><td>
</td></tr>
<tr><td>Connect to Review Server</td><td>
</td></tr>
<tr><td>Enable Stereo view</td><td>
</td></tr>
<tr><td>Start Tracking</td><td>
</td></tr>
</table>

Table 3.2: Sequence diagram by use case

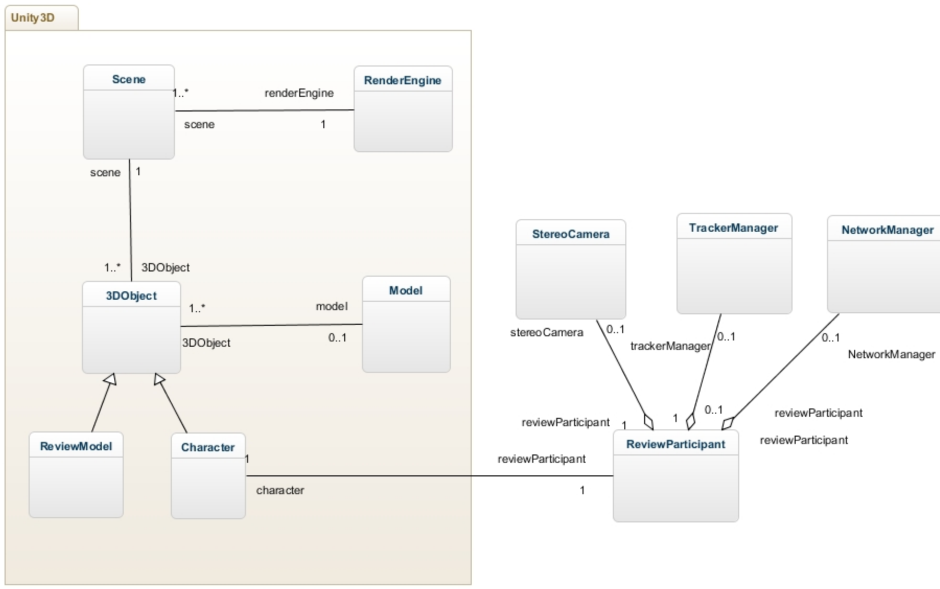


Figure 3.2: High Level Class Diagram

3.8 Implementation

So far, a review of how to implement the application has been given. This section presents a detailed explanation about the construction process and its limitations.

3.8.1 Working with Unity 3D

Unity 3D is a content creation software, as such it provides a development environment that differs from the most commonly used integrated development environments (IDEs) in the software industry. One of the main differences is that it is though to visually design a scene first and then program the different interactions within the 3D objects. This approach is similar to some CAD applications graphical interface such as Autodesk's Revit, where architects and engineers create different models while having visual feedback of the changes they make. Figure 3.3 shows the initial screen in Unity 3D.

From left to right, the first tab called "Hierarchy" holds the objects contained within a 3D scene.

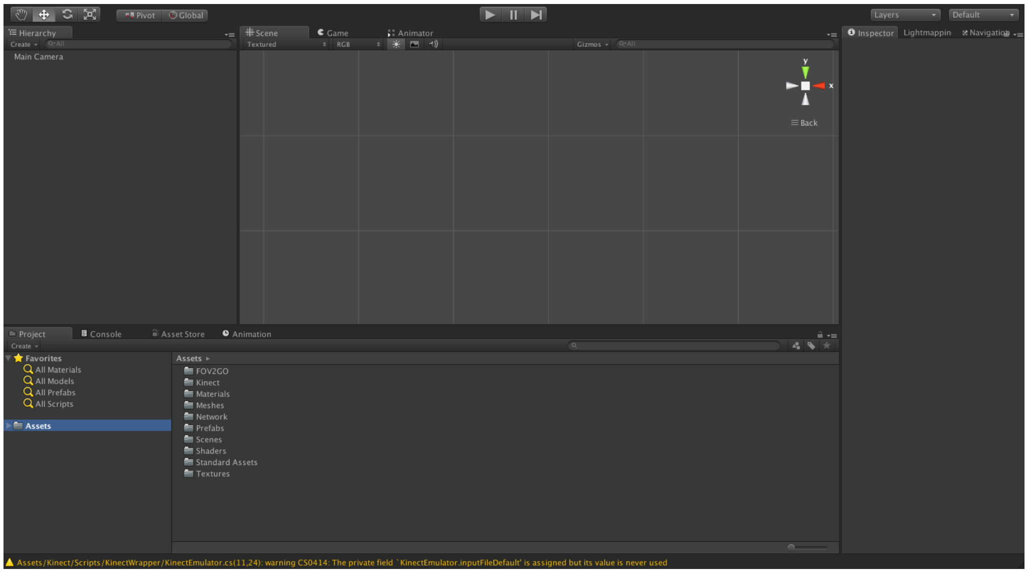


Figure 3.3: Unity 3D initial screen

In Unity 3D the objects are called game objects, they can contain lights, 3D models, text elements, among others. A game object can have several game objects under it, the top game object would be parent of the low level objects, hence the name hierarchy view. Any transformation on the parent object affect all the children objects.

The second tab called "Scene" shows the visual representation of the game objects and its position inside the virtual world. This view can be think of as the sandbox where a designer can place the 3D objects that compose a virtual environment and see how they would look before creating the software build.

The third tab called "Inspector" shows the components of a game object. At first, all game objects share a component called "Transform", this component holds the position, rotation and scale of the game object. From there, a game object can contain a variety of different components, some predefined such as cameras, audio listeners, text elements, and some user made in the form of scripts.

The bottom side of figure 3.3 shows a tab called "Project". This view holds all the resources available to use within a scene. 3D models, textures, animations, scripts, prefabs, and others. Prefabs are user created game objects, e.g. a special kind of camera or playable character.

Unity scripts can be written in three different programming languages: C#, UnityScript -

javascript like syntax -, and boo. Scripts play the most important feature of Unity 3D, with them a programmer can change the behavior of a game object, manage a group of game objects, and even create new game objects in the 3D scene. Unity comes with Mono Develop IDE to write the scripts, but a script can be written in any text editor, like Microsoft's notepad. Figure 3.4 shows Mono Develop user interface.

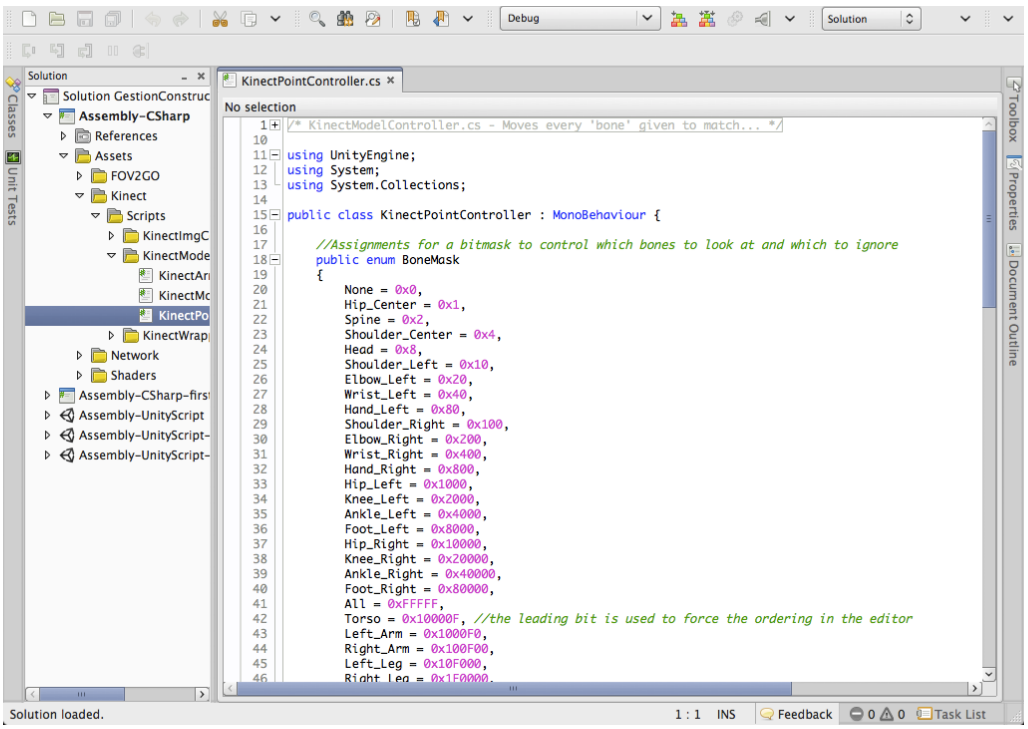


Figure 3.4: Mono Develop user interface

3.8.2 Game objects and prefabs

For the development of this application the following game objects were created in the project:

- NetViewCube: The purpose of this game object is to implement network functionality. This game object has a NetworkView component, which is the Unity 3D predefined component that implements Remote Procedure Calls and state synchronization mechanisms, and a network script, which is the class that handles the network interactions created.
- MainCamera: The purpose of this game object is to be the window of what the user sees.

It has the same components as the predefined Unity 3D camera plus a camera script from the FOV2GO package that transforms a this game object into a side by side stereo array of cameras.

- skybox3D: The purpose of this game object is to give the illusion of being under the sky in the scene. It is needed because predefined skybox object from Unity doesn't work with FOV2GO stereo solution. This game object is part of FOV2GO package.

- KinectManager: The purpose of this game object is to hold the Kinect SDK wrapper from Carnegie Mellon, this gives an interface to use the Kinect data from within Unity as if it was a regular Windows SDK development.

The other game objects are replaceable by the modeller who wants to prepare a review meeting.

As in for the prefabs, one under the name on FPSEntity was created. This prefab contains a 3D model of the character that is going to be visible by others, a FPSEntity model script that contains the actions it can perform such as move forward, backwards, jump; a Kinect script in charge of processing the data coming from the kinect and a light that will work as a pointer to other objects in the scene.

3.8.3 Networking implementation

The network script is the one responsible to handle all the network interactions within the application. This implementation is a non authoritative implementation of a server and client. This choice was made because the activity to be performed is the review of a design, making the changes of the review is a complex activity and it is out of the scope of the meeting, hence out of scope of the application. It is based on RPC instead of state synchronization to give a more fluid sense of movement of the non local review participants. Figure 3.5 shows the fields and methods of this class.

As stated before, anyone running the program can act as the host of the meeting, or as an attendee. This functionality is handled on the OnGUI method. OnGUI is a predefined method of a GUI component in Unity, this method is called every frame and it's purpose is to draw on screen Graphical User Interface (GUI) objects, such as buttons, labels, checkboxes, etc. As the start server and connect to a server are methods of the NetworkView component, the OnGUI method creates

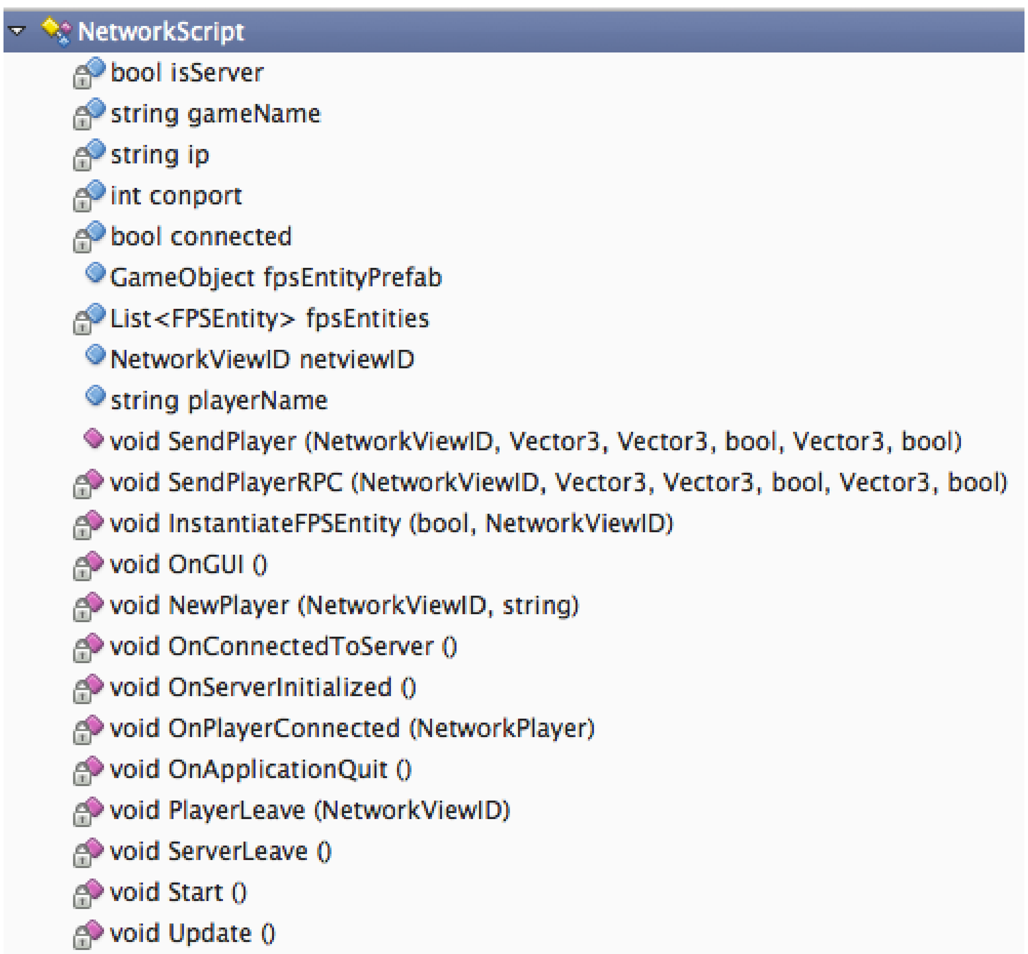


Figure 3.5: NetworkingScript class fields and methods.

buttons to start the server and connect to a server and these network methods are called when this buttons are pressed. After checking that a connection is possible, InstantiateFPSEntity method is called, this creates an instance of a character within the world and updates a list of users connected to the server.

SendPlayer is the method called from the FPSEntity to update character information across the network, while SendPlayerRPC is the method used by the NetworkScript to all connected clients.

3.8.4 Tracking system implementation

Using the tracking system is simple, after setting up the wrapper in the KinectManager game object all that is needed is to create a script to use the information provided by the KinectManager. In this implementation a script called KinectArmsController was attached to the FPSEntity prefab in its root to control the movements in X axis and to its camera to control its movements in Y axis.

KinectArmsController script first polls if tracking system is needed, in order to avoid having the user always pointing at something within the 3D world. Then, if it is tracking, polls the wrapper for the tracking information, as one would do using the Microsoft's Kinect SDK. Finally the script translates the tracking info into movement of the camera within the application. The joint being tracked in this environment is the wrist because hand tracking gives less accurate information unless the user is really near the Kinect hardware.

3.8.5 Usage

The main flow of a review meeting with the developed tool is the following: On a regular basis the project manager schedules a meeting with the different specialist to review the design. Then the modellers set up the models in Unity by dragging the model to the project view and then dragging the resulting game object into the hierarchy or scene view. After that, modellers position the game object models and lighting using the inspector and scene view. When the set up is complete, modellers build the project and make it available over the internet, or as downloadable binary and notify all the review participants. Before the meeting takes place, the participants log in to the website where the application is hosted or download the binary application. At the time of the meeting the attendees should connect to the host application providing the information sent by modellers, either look up in the master server if available or using direct connection with IP and port information. Using a conference system like Vidyo, Skype or google's hangout combined with the review application the review meeting starts. With the review system facilities like pointing and taking screenshots, participants agree on the changes that need to be made in the design. When the meeting is over participants share their screenshots and notes of the conference using email or another sharing system defined for the project, and the process iterates until the design phase is finished. Figure 3.6 shows an example view of the built application.

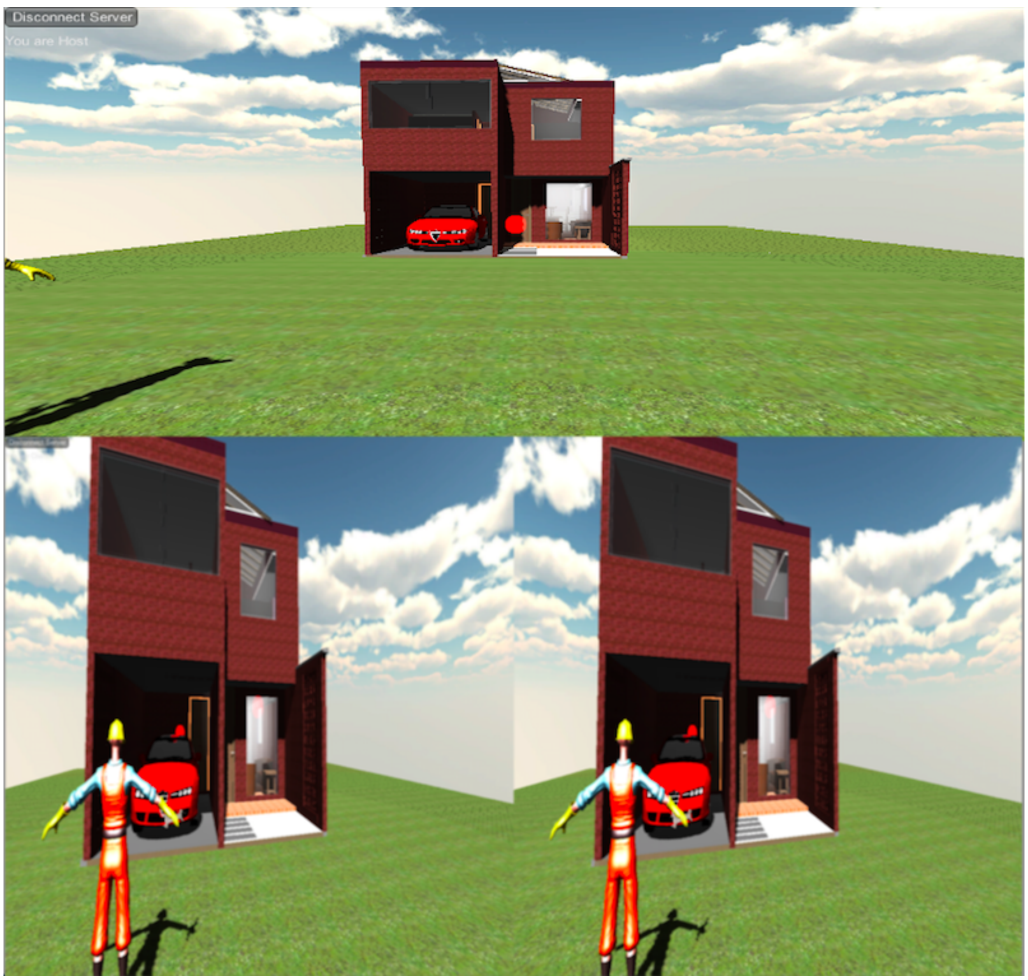


Figure 3.6: Top - Host view from a regular screen. Bottom - Client view with stereoscopic setting on.

Chapter 4

Experiments

In this chapter a detailed explanation of the conducted experiments of the tool is presented. Two major experiments took place, the first one compared the revision of a design in the form of 2D plans against a 3D model, the second one compared a collaborative revision of 2D plans against a collaborative revision of 3D models. Both of the experiments had a perception survey and an examination of the performance on the task. This chapter ends with the results of the experiments.

4.1 Development

The experiments intended to demonstrate the following hypothesis:

A1: More errors are detected while reviewing BIM models than in the traditional plans review.

A0: There is no difference in the number of errors found in the review of BIM models and traditional plans.

B1: More errors are detected with collaborative review of BIM models than collaborative review of traditional plans.

B0: There is no difference between the number of errors found on the collaborative review of BIM models and collaborative review of traditional plans.

4.1.1 Experiments setup

Experiment A setup: This experiment's population were civil engineer students who were in approximately half of their career, between 4th and 5th semester students. The students were asked conduct a review of a BIM model and the architectural plans from the same project. Before starting the experiment attendees were separated into groups of 2 people. For each two groups, one was located in the virtual reality room of EAFIT University and reviewed the 3D model. The other group sat in a separate room and reviewed the plans from the same project. The model and the plans intentionally had several architectural errors that the participants had to discover. Each group had five minutes to perform their respective tasks. Once time ended, participants filled a survey to assess their perception of the experience.

Experiment B setup: For this experiment, civil engineer students who were between 4th and 5th semester were scheduled. Before starting the experiment attendees were divided into groups of 2 people. For each two groups one collaboratively reviewed the 3D model, a one member in the virtual reality room and the other one in a separate room, The other group was also separated into 2 rooms with a computer each and performed a collaborative review of the plan from the same project. The model and the plans intentionally had several architectural errors that the participants had to discover. Each group had five minutes to perform their respective tasks. Once time ended, participants filled a survey to assess their perception of the experience.

Table 4.1 shows the variables of the two experiments.

	Experiment A	Experiment B
Independent Variables	Review method (BIM or traditional).	Collaborative review method (BIM or traditional).
Dependent Variables	Number of errors found in the test.	Number of errors found in the test.
Control Variables	Time to complete the test.	Time to complete the test.

Table 4.1: Experiment variables

4.1.2 Physical setup

For this experiments the following equipment was used:

Virtual reality room EAFIT University:

- LG 3D DLP projector
- Dell Precision T7600 Computer
  - Intel Xeon Processor E5 - 2667 0@2.9 GHz
  - Video card Nvidia Quadro 6000 (6 GB)
  - 16 GB of RAM
  - Microsoft Windows 7 - 64 bit Operating System
- Microsoft Kinect for Xbox 360
- Xbox 360 Wireless Controller
- Review application created in Unity 3D
- Silver screen.

Meeting Room 1:

- HP Computer
  - Intel Core i7 870@2.93 GHz
  - Video Card ATI Radeon HD 4600 (512 MB)
  - 16 GB of RAM
  - Microsoft Windows 7 - 64 bit Operating System
- Xbox 360 Wireless Controller
- Review application created in Unity 3D

Meeting Room 2:

- iMac Computer:
  - Processor Intel Core 2 GHz Duo@3.06
  - Video Card ATI Radeon HD 4670 (512 MB)
  - 4 GB of RAM

  - OS X 10.6.8

- MacBook Air:

  - Processor Intel Core i5@1.8 GHz
  - Video Card Intel HD Graphics 4000 (1 GB)
  - 4 GB of RAM
  - OSX 10.9.1

In total 28 people attended. 14 participated in the test without collaboration (Test A) and 14 participated in the collaborative test (Test B). The average population age was 22 years old.

4.2 Results

This section shows the results of the conducted experiments.

4.2.1 Perception results

To create the user perception survey the Technology Acceptance Model (TAM) was used [42]. This model aims to identify the perceived usefulness (PU) and perceived ease of use (PEOU) of certain technology, also known as the PUEU model for studies of human computer interface usability. The model consists of 12 questions, 6 for measuring PU and 6 for PEOU measures. In this survey only 11 questions of TAM were used due to the similarity between two of the questions proposed by the author. This model was chosen as an ideal research tool to measure the perception because it is not tied to a specific development but rather the perception of the user from the various tools used during the experiment.

The model was translated and adapted to the specific context of the test participants for better understanding of the activity. The survey presented five possible answers between 5 and 1, being 5 where the participant fully agreed to the statement that was made, and 1 if the participant strongly disagree. Table 4.2 on page 46 shows the survey results for experiment A. Table 4.3 on page 47 shows the survey results for experiment B.

Experiment A

From the surveyed students from experiment A, 71% answered that they completely agree with the statement "immersive virtual reality will make it easier for me to understand a class on design review", this gives the author a hint about the high acceptance of the tool. This also provides hints about the potential it has for performing this activity. Regarding the perceived ease of use, 86% of respondents completely agree that interaction with the tool is easy. This is possibly related with the age range of participants.

	Statement	Completely agree	Partly agree	Neither agree nor disagree	Partly disagree	Completely disagree
1	The use of immersive virtual reality in class would allow me to perform review tasks more quickly	71%	29%	-	-	-
2	The use of immersive virtual reality could improve my performance in class’	71%	14%	14%	-	-
3	The use of immersive virtual reality will improve my effectiveness in class	57%	29%	14%	-	-
4	The use of immersive virtual reality will make it easier for me to understand a class on design review	57%	43%	-	-	-
5	The use of immersive virtual reality will be helpful during class’œ or activity	72%	23%	-	-	-
6	Learn how to use this virtual reality tool will be easy for me	57%	29%	14%	-	-
7	It will be easy for me to make the tool do what I want it to do	43%	43%	14%	-	-
8	My interaction with the virtual reality tool was clear and comprehensible	86%	-	14%	-	-
9	Interacting with the virtual reality tool was flexible	57%	29%	14%	-	-

10	It would be easy for me to master this virtual reality tool	57%	29%	14%	-	-
11	I found the immersive virtual reality tool easy to use	71%	14%	14%	-	-

Table 4.2: Perception Results Experiment A

Experiment B

Experiment B respondents showed that, 70% perceived the collaborative tool as a mean to perform review tasks faster. As each team member had its own view of the model being revised, the author believes users can perceive that as the task being performed more quickly. Also , 82% of respondents completely agree that using the tool will be useful to understand a class on the subject. This is a hint on how seeing a 3D model can increase the awareness of errors where traditional plans make it difficult. Respondents also reported that the tool will be helpful for the task at hand, with 72% responding that they completely agree.

	Statement	Completely agree	Partly agree	Neither agree nor disagree	Partly disagree	Completely disagree
1	The use of immersive virtual reality in class would allow me to perform review tasks more quickly	70%	25%	5%	-	-
2	The use of immersive virtual reality could improve my performance in class'	63%	34%	2%	1%	-
3	The use of immersive virtual reality will improve my effectiveness in class	64%	31%	5%	-	-
4	The use of immersive virtual reality will make it easier for me to understand a class on design review	82%	18%	-	-	-
5	The use of immersive virtual reality will be helpful during class'œ or activity	72%	23%	5%	-	-
6	Learn how to use this virtual reality tool will be easy for me	64%	27%	8%	1%	-

7	It will be easy for me to make the tool do what I want it to do	48%	41%	11%	-	-
8	My interaction with the virtual reality tool was clear and comprehensible	70%	27%	2%	1%	-
9	Interacting with the virtual reality tool was flexible	67%	30%	1%	1%	-
10	It would be easy for me to master this virtual reality tool	63%	34%	4%	-	-
11	I found the immersive virtual reality tool easy to use	60%	30%	10%	-	-

Table 4.3: Perception Results Experiment B

4.2.2 Performance results

For the performance results of the task, independent t-tests were conducted for experiment A and experiment B respectively. Table 4.4 on page 47 and table 4.5 on page 49 shows the results of both test. R statistical program was used to perform this comparison.

Experiment A

Initial hypothesis for experiment A stated that subjects using the BIM model could find more errors than those using 2D plans. The statistical evaluation of the results proved that this intuition was correct. With 4.5 degrees of freedom the results of a Welch two sample t-test shows that $p - value = 0.00035$, hence the alternative hypothesis can be accepted with a 95% confidence. Table 4.4 shows a summary of the t-test results. This gives an indication that using virtual reality for a review process is a better method than reviewing 2D plans.

Welch Two Sample t-test		
data	Model Errors and Plans Errors	
t = 5.5549	df = 4.523	p-value = 0.003552
alternative hypothesis:	true difference in means is not equal to 0	
95 percent confidence interval:	1.566522	4.433478
sample estimates:		
Mean of Model Errors	Mean of Plans Errors	
4.75	1.75	

Table 4.4: Welch Two Sample t-test for Experiment A

A graphical way to demonstrate that the two samples have different means, also supporting the results from the t-test, is presented in the box-and-whisker plot figure 4.1 on page 48. Here it is clearly seen that non of the samples means are overlapping.

Figure 4.1: Box and whiskers plot experiment A

These results show that there are reasons to believe that reviewing process in the AEC industry should include the construction and refinement of 3D models. This will improve the accuracy of the design in terms of lesser errors and will help a better understanding within the different disciplines views involved in this process.

Experiment B

For experiment B the hypothesis was that collaborative review of 3D models was better than collaborative review of 2D plans. In this case the Welch two sample t-test showed a $p - value = 0.0454$. This value still lesser than 0.05 and the alternative hypothesis can be accepted with 95% of confidence. Table 4.5 shows the summary of the t-test results. Although difference this time is not as big as the one with experiment A, this results gives an indication that collaborative virtual reality is a fit alternative to perform geographically dispersed review of AEC designs.

Welch Two Sample t-test		
data	Collaborative Model Errors and Collaborative Plans Errors	
t = 2.8284	df = 4.154	p-value = 0.0454
alternative hypothesis:	true difference in means is not equal to 0	
95 percent confidence interval:	0.06510347	3.93489653
sample estimates:		
Mean of Collaborative Model Errors	Mean of Collaborative Plans Errors	
2.5	0.5	

Table 4.5: Welch Two Sample t-test for experiment B

Supporting the previous results, a box-and-whisker plot shown in figure 4.2 on page 49 is presented. This plot graphically shows that the means between the two samples are different by not overlapping.

Figure 4.2: Box and whiskers plot experiment B

With experiment B it is seen that collaborative virtual environments can support the review design process. This can be helpful in geographically separate design teams where physical meetings are not possible due to cost or time restrictions. More runs of this experiment have to be done to improve the t-test results and be more confident with the acceptance of the hypothesis.

Chapter 5

Conclusion

The review design phase of a project is one of the most important processes in the AEC industry. Flaw designs lead to over cost in budget and construction time leaving stakeholders unhappy about the final product. In examined researches, virtual reality has proven to be an effective tool to improve design challenges, leveling expectations from clients and allowing specialist to have visual feedback on their work. The present document gave an explanation on how to develop a collaborative review virtual environment, improving review process time allowing specialist to be in different geographical locations.

During the experiments in general, all participants showed excitement at the time of using the the review process application. Control group participants found the time to execute the task really limited. In several cases no errors in the plans were found by these participants. By contrast, while reviewing the model at least two errors were found by the participants. It is sensed that having a 3D representation of an AEC design its a more natural mean to perform a review task, compared against 2D plans review.

Several technical difficulties were found while performing the experiments, such as tracking system location within the virtual reality lab. Since the Kinect sensor requires that people stand right in front of it, it was uncomfortable for the participants to use it the way the room had it set up, next to the screen. This recommendations were stated on the user perception survey. Better work in setting up the experiment lab needs to be done for future experimentations.

Finally, the results from the experiments show that there are reasons to believe that reviewing process in the AEC industry should include the construction and refinement of 3D models. This will improve the accuracy of the design in terms of lesser errors and will help a better understanding within the different disciplines views involved in this process. Also collaborative virtual environments can support the review design process. This can be helpful in geographically separate design teams where physical meetings are not possible due to cost or time restrictions. More runs of this experiment have to be done to improve the t-test results and be more confident with the acceptance of the hypothesis.

5.1 Future Work

To improve this application the following developments can be done:

- Provide a mechanism to name the characters within the virtual environment.
- Add an animated version of the characters used by the review participants.
- Provide a new controller with free camera movement around all the environment.
- Integrate the sharing system within the application to avoid the use of other systems outside the application.
- Integrate a communication system within the application to avoid the use of other systems outside the application.
- Create a versioning system of review meetings where review participants can see the history of reviews.
- Add the possibility to view the cameras of other participants to know exactly what they are looking at.
- Create a model import facility within the application which provides the same functionality as Unity 3D of placing the model to be reviewed, and to add lighting and the textures to the environment.

A lot of studies can be made from the previous application improvements. E.g, adding animation to the characters, can be a research about the influence of animation within a collaborative virtual environments; the creation of a sharing system within a 3D environment can be a research itself that deals with concurrent use of information; creating the model import facility can be a research about the different 3D model representation standards and which one would serve better to collaboratively build a 3D world.

The author is aware that other implementations can be made to solve this problem, further study on this subject can be made.

To finish, the developed tool could also be used in other review processes outside the AEC industry, like in product design, other studies need to be done to prove this statement.

Bibliography

[1] A. Salman, "Building Information Modeling (BIM): Trends, Benefits, Risks, and Challenges for the AEC Industry," *Leadership and Management in Engineering*, vol. 11, no. 3, pp. 241–252, 2011.

[2] H. El Bibany, "AEC Collaborative Information Systems: From Requirements to Architecture," *Automation and Robotics in Construction Xi*, pp. 515–522, 1994.

[3] A. Mokhtar, C. Bedard, and P. Fazio, "Information Model for Managing Design Changes in a Collaborative Environment," *Journal of Computing in Civil Engineering*, vol. 12, no. 2, pp. 82–92, Apr. 1998.

[4] R. Fruchter, "A/E/C Teamwork: A Collaborative Design and Learning Space," *Journal of Computing in Civil Engineering*, vol. 13, no. 4, pp. 261–269, Oct. 1999.

[5] M. Rosenman and F. Wang, "CADOM: A Component Agent based Design Oriented Model for Collaborative Design," *Research in Engineering Design*, vol. 11, no. 4, pp. 193–205, Dec. 1999.

[6] C. Anumba, O. Ugwu, L. Newnham, and A. Thorpe, "Collaborative design of structures using intelligent agents," *Automation in Construction*, vol. 11, no. 1, pp. 89–103, Jan. 2002.

[7] F. Pena-Mora and G. H. Dwivedi, "Multiple Device Collaborative and Real Time Analysis System for Project Management in Civil Engineering," *Journal of Computing in Civil Engineering*, vol. 16, no. 1, pp. 23–38, Jan. 2002.

[8] A. C. B. Garcia, J. Kunz, M. Ekstrom, and A. Kiviniemi, “Building a project ontology with extreme collaboration and virtual design and construction,” *Advanced Engineering Informatics*, vol. 18, no. 2, pp. 71–83, Apr. 2004.

[9] P.-H. Chen, L. Cui, C. Wan, Q. Yang, S. K. Ting, and R. L. Tiong, “Implementation of IFC-based web server for collaborative building design between architects and structural engineers,” *Automation in Construction*, vol. 14, no. 1, pp. 115–128, Jan. 2005.

[10] D. J. Nicol and I. A. MacLeod, “Using a shared workspace and wireless laptops to improve collaborative project learning in an engineering design class,” *Computers & Education*, vol. 44, no. 4, pp. 459–475, May 2005.

[11] W. D. Li and Z. M. Qiu, “State of the art technologies and methodologies for collaborative product development systems,” *International Journal of Production Research*, vol. 44, no. 13, pp. 2525–2559, Jul. 2006.

[12] CAMACOL, “Colombia Construccion Informe 2013,” CAMACOL, Tech. Rep., 2013.

[13] A. B. CRAIG, W. R. SHERMAN, and J. D. WILL, *Developing virtual reality applications: Foundations of effective design.* Boston: Morgan Kaufmann, 2009.

[14] T. G. ZIMMERMAN and T. Z. LANIER, “Computer data entry and manipulation apparatus and method,” 1991.

[15] M. RAMOS, J. DELGADO, D. CERVANTES, and R. LERICHE, “Creacion de ambientes virtuales inmersivos con software libre,” *Revista Digital Universitaria*, vol. 8, no. 6, 2007.

[16] Microsoft Corporation, “Getting Started with DirectX Graphics (Windows).” [Online]. Available: http://msdn.microsoft.com/en-us/library/windows/desktop/hh309467(v=vs.85).aspx

[17] OpenGL Community, “FAQ - OpenGL.org.” [Online]. Available: http://www.opengl.org/wiki/FAQ#What_is_OpenGL.3F

[18] EON reality, “EON Studio | Build Interactive 3D Content Quickly and Easily.” [Online]. Available: http://www.eonreality.com/eon-studio

[19] EPIC Games Inc, “Free Game Engine for Indie Game Development | UDK Unreal Developers Kit.” [Online]. Available: http://www.unrealengine.com/en/udk/

[20] ——, “UDK Licensing Unreal Technology.” [Online]. Available: https://www.unrealengine.com/udk/licensing/purchase/

[21] CryTek, “Crytek | MyCryENGINE.” [Online]. Available: http://mycryengine.com/index.php?conid=59

[22] Unity Technologies, “Unity Game engine, tools and multiplatform.” [Online]. Available: http://unity3d.com/unity/

[23] A. S. Tanenbaum and D. J. Wetherall, “Introduction,” in *Computer networks*, 5th ed. Boston: Prentice Hall; 5 edition (October 7, 2010), 2010, p. 2.

[24] Unity Technologies, “Unity High Level Networking Concepts.” [Online]. Available: http://docs.unity3d.com/Documentation/Components/net-HighLevelOverview.html

[25] G. BURDEA, P. RICHARD, and P. COIFFET, “Multimodal virtual reality: Input output devices, system integration, and human factors,” *International Journal of Human Computer Interaction*, vol. 1, no. 8, pp. 5–24, 1996.

[26] NaturalPoint Inc., “OptiTrack - V100:R2 - Technical specifications for the V100:R2.” [Online]. Available: http://www.naturalpoint.com/optitrack/products/v100-r2/specs.html

[27] University of North Carolina at Chapel Hill, “VRPN.” [Online]. Available: http://www.cs.unc.edu/Research/vrpn/

[28] Im in VR, “MiddleVR for Unity.” [Online]. Available: http://www.imin-vr.com/middlevr-for-unity/

[29] Worcester Polytechnic Institute, “Unity Indie VRPN Adapter (UIVA).” [Online]. Available: http://web.cs.wpi.edu/ gogo/hive/UIVA/

[30] Microsoft Corporation, “Microsoft Kinect Documentation - Sensor Components and Specifications.” [Online]. Available: http://msdn.microsoft.com/en-us/library/jj131033.aspx

[31] ——, "Microsoft Kinect Documentation - Skeletal Tracking." [Online]. Available: http://msdn.microsoft.com/en-us/library/hh973074.aspx

[32] University of Southern California Institute for Creative Technologies, "Flexible Action and Articulated Skeleton Toolkit (FAAST)." [Online]. Available: http://projects.ict.usc.edu/mxr/faast/

[33] Motion Arcade Inc, "Zigfu Legacy." [Online]. Available: http://zigfu.com/en/downloads/legacy/

[34] Carnegie Mellon University Entertaintment Technology Center, "Microsoft Kinect - Microsoft SDK - Unity3D." [Online]. Available: http://wiki.etc.cmu.edu/unity3d/index.php/Microsoft_Kinect_-_Microsoft_SDK

[35] WorldViz, "Anaglyphic (red/cyan glases)." [Online]. Available: http://docs.worldviz.com/vizard/Anaglyphic_(red_green_glases).htm

[36] P. Bourke, "Create side-by-side stereo pairs in the Unity game engine," 2008.

[37] The Theoretical and Computational Biophysics Group, "Side By Side and Cross Eyed Stereo." [Online]. Available: http://www.ks.uiuc.edu/Research/vmd/vmd-1.9.1/ug/node102.html

[38] WorldViz, "Frame sequential." [Online]. Available: http://docs.worldviz.com/vizard/Frame_sequential.htm

[39] The Theoretical and Computational Biophysics Group, "Quad-buffered Stereo." [Online]. Available: http://www.ks.uiuc.edu/Research/vmd/vmd-1.9.1/ug/node101.html

[40] Nvidia Corp, "3D Vision Pro FAQ | NVIDIA." [Online]. Available: http://www.nvidia.com/object/3dvision-pro-faq.html

[41] University of Southern California Institute for Creative Technologies, "FOV2GO Developer." [Online]. Available: http://projects.ict.usc.edu/mxr/diy/fov2go-developer/

[42] F. D. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology," *MIS quarterly*, vol. 13, no. 3, pp. 319–340, Dec. 1989.