Int J Interact Des Manuf (2017) 11:931–947
DOI 10.1007/s12008-016-0301-z

TECHNICAL PAPER

Collaborative design model review in the AEC industry

Juan Gonzalo Cárcamo[1] · Helmuth Trefftz[1] · Diego Andrés Acosta[1] · Luis Fernando Botero[1]

Received: 2 September 2015 / Accepted: 5 January 2016 / Published online: 16 January 2016

Abstract The end result of this research is an application created using freely available tools applied to a case scenario to perform review meetings by different specialists in real time. Initial tests on the system have been made with Civil Engineering students showing that this virtual reality tool eases the burden of performing engineering drawings reviews that are traditionally done at the same geographical space.

Keywords AEC · Collaborative virtual environments · Collaborative design

1 Introduction

The construction industry in Colombia generates a high impact on Colombian's economy, 9 % contribution to GDP is estimated to related sectors and occupation of about 6 % of the employed population [3]. However, construction underperforms compared to other industries in product quality, industrial safety, environmental impact and meeting deadlines. Research projects in the field have shown the flaws influence the design process and project planning, which have a negative impact on the performance of the industry, a situation generated by the growing complexity of projects, reduced execution times, and informality of the production system.

Building Information Modeling (BIM) and Virtual Reality models can contribute to the understanding of construction projects, facilitating their planning and execution, improving project performance, by helping users detecting inconsistencies in pre-construction phases of the construction sequence, or detecting flaws in the interferences between different specialists designs and improving visualization, allowing for a correct implementation of the project.

As a result of the project, being advanced by research groups in Virtual Reality and Construction Management at EAFIT University, since the late 2010, the infrastructure for virtual modeling and visualization in virtual environments on a large format projection screen became available. This work intends to show how using currently available infrastructure, 3D modeling capabilities and collaborative work among specialties of Civil Engineering, Architecture and / or different research groups or companies can improve the design process in AEC industry.

2 Background

The term "Virtual Reality" can be traced to a literary work of 1938 ("The Theater 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. 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.

✉ Juan Gonzalo Cárcamo
jcarcamo@eafit.edu.co

Helmuth Trefftz
htrefftz@eafit.edu.co

Diego Andrés Acosta
dacostam@eafit.edu.co

Luis Fernando Botero
lfbotero@eafit.edu.co

1 Carrera 49 N 7 Sur-50, Medellín, Colombia

When a user interacts in a Virtual Reality 3D world, the set of emotions and feelings that the person experiences, such as surround sound, 3D display of objects, among other experiences, is what allows for a complete sense of immersion in the 3D virtual reality world.

One of the main benefits of using Virtual Reality is that it 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.

2.1 Technical considerations of virtual reality systems

Typically Virtual Reality applications require the following items for its construction [20]:

- *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 of a predefined path, move an object in the 3D scene without modifying the user's position, moving user's rotation, among others.
- *Collision handling* To increase the immersion within the 3D scene, computer programs that handle user restrictions against objects within the virtual environment should be built. e.g user's avatar collision with the floor and walls objects to give the impression of walking in the scene or perform some action such as opening a door once the user is in front of it.
- *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 behaviours, 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.

2.2 Building information modelling

Building information Modelling can be described as the mix of technologies and a methodology to improve the Architecture, Engineering and Construction (AEC) product development by reducing project delivery time, increasing productivity and quality, and decreasing project cost [21].

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's stakeholders to have a single, unified view of the project. This improves the collaboration among the different specialties that interact in this kind of initiatives.

2.3 Collaborative environments and interactive design

Interactive design is a mixture between modelling techniques, with the goal to center the design process on humans [11]. Technology plays a very important part on this. If it wasn't for emerging CAD technologies it would be very difficult to integrate all different engineering know-how into one single design process. Having a single process means to provide tools to enable the sharing of knowledge hence the need of developing a collaborative environment capable of allowing all stakeholders to perform their jobs according to their knowledge but aware of the work of others. For the Architecture, Civil Engineering, and Construction Industry, building design has had 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.

In order to create an effective collaborative environment, El-Bibany proposes the following capabilities of the system [7]:

– 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.

Of course all this capabilities could be met without using technology, but it is the intention of the authors to show how using a virtual reality environment could greatly improve the decision making in building design, and by allowing different disciplines such as architecture and engineering looking at the same BIM model at the same time even when located at a different geographical position the sensorial approach to interactive design is used.

3 Contribution

A detailed explanation on how the proposed system was built is given here. First it starts with the main requirements of the desired visualization-based collaborative system, and then it gives the reader an analysis of the tools considered to achieve this goal. Finally it concludes with a use case scenario 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 the "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 Sect. 2 a lot of advances have been developed in the Computer Graphics field. While doing this project the authors 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, a brief summary of the tested tools is presented.

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 [13]. 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 [19].

For the purpose of this work these two APIs were compared against the requirements. The first drawback was that these APIs only deal with graphics rendering, meaning that every other functionality in the requirements needs to be done through other APIs. Also 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 article.

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 [8]. 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 licensed 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 [9] used in hundreds of AAA quality games. It has all the functionality of Unreal Engine without the full access to its source code [10]. 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 with texturing 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 [5]. 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 [27]. 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. Table 1, on page 9, 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 proposals 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.

How to handle communications in Networked Virtual Environments is an open question. There are many possible protocols that can be chosen, depending on how much synchronization or how much interactivity is required by the application [22]. Furthermore, in the presence of heterogeneity in the environment (including hardware, network conditions, etc.), it is necessary to device a mechanism to cope with it without affecting the collaborative task [26].

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) [28]. 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 servers where clients are responsible

Table 1 Readiness analysis summary

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

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 users do not perceive 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 gets 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 [2].

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 authors, 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 [17]. 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 [29].

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 [12]. 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 [32] 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 [14]. 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 [15].

Three approaches were found to use Kinect with Unity 3D, using FAAST VRPN server [30] with UIVA [32], which is already supported by this solution, Using OpenNI driver and open source wrapper developed by Zigfu [16] and Carnegie Mellon University Entertaintment Technology Center Microsoft's SDK wrapper [4].

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 [33].

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 [1], 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 [25], 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 1920×1080 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 [34]. 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 [24].

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 [18]. 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 [1]. 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 [31]. 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 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 a static representation of the types of objects being modelled. This representation shows the associations between the different types, and in further refine-

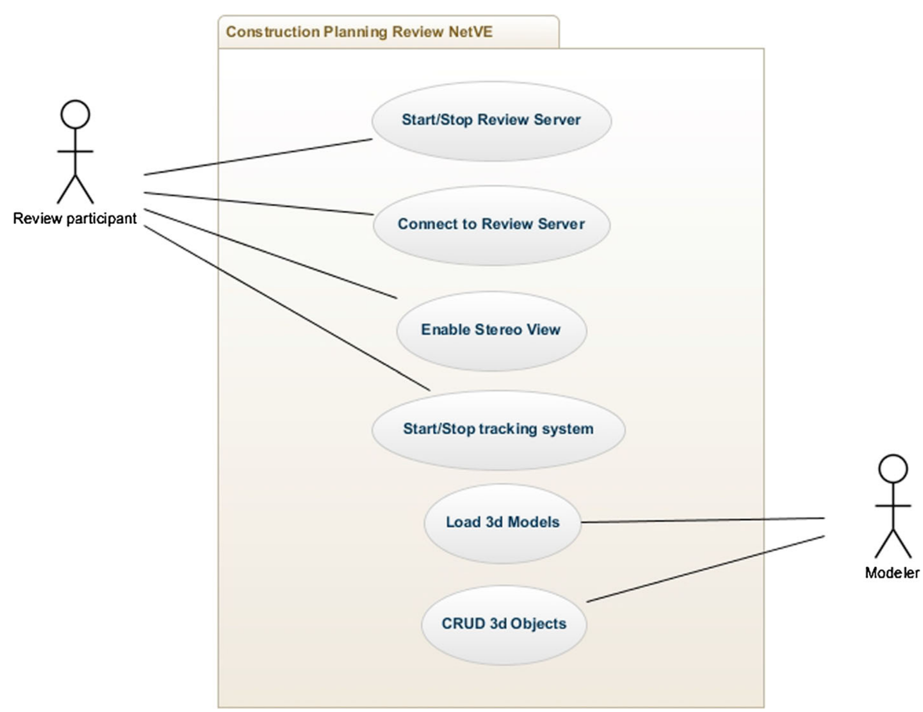


Fig. 1 High level use case diagram

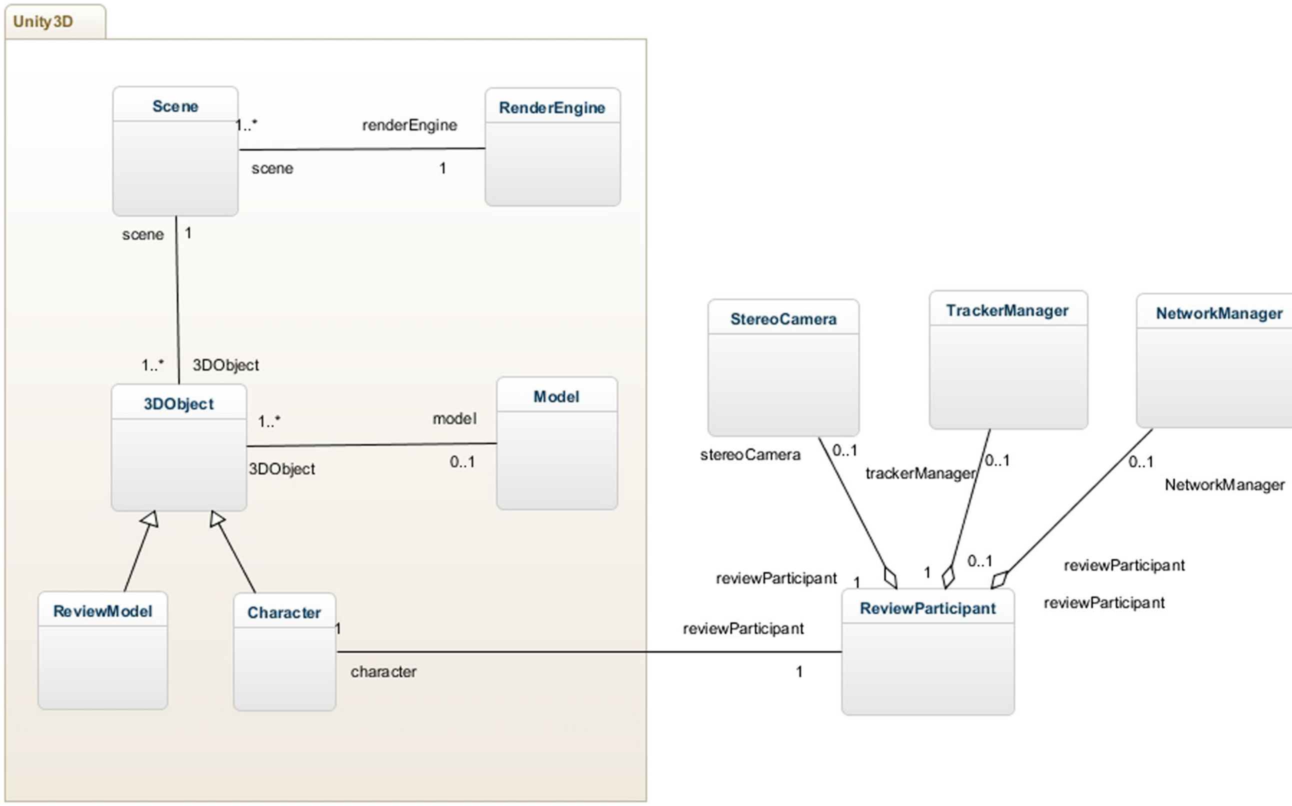


Fig. 2 High level class diagram

ments contain each classes' attributes and actions, to latter be programmed as a fields and methods according to the Object Oriented programming paradigm. Figure 2 shows the high level class design for the proposed 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 authors' 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 2 on page 16 shows the main behaviours in the system ordered by the use case they support.

3.8 Implementation

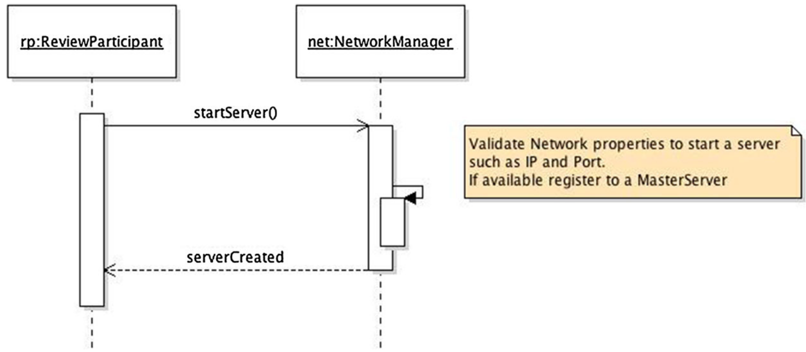
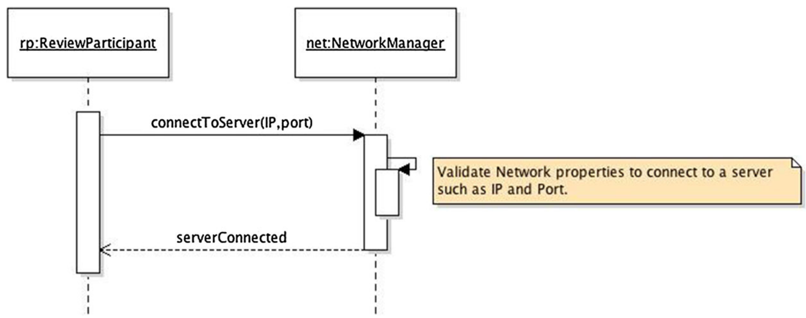
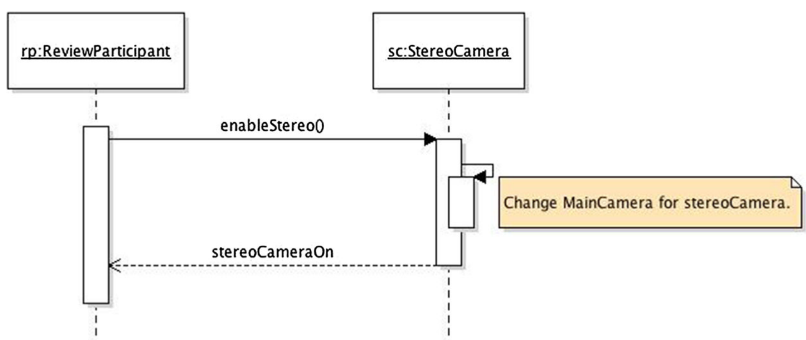
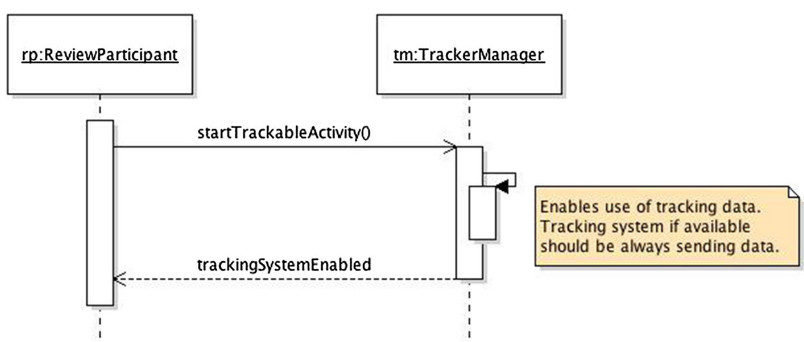
So far, a review of how to implement the application has been given. This section presents a detailed explanation about the construction process and its limitations.

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 shows the initial screen in Unity 3D.

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

Table 2 Sequence diagram by use case

<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>

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 thought 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 compo-

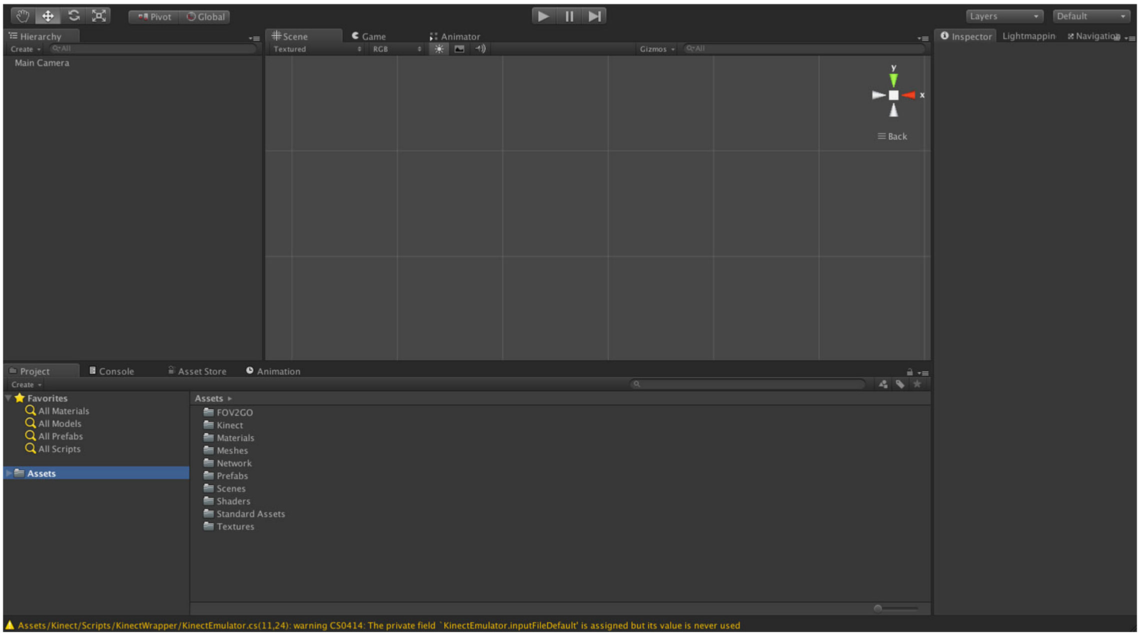


Fig. 3 Unity 3D initial screen

nent 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 Fig. 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 4 shows 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

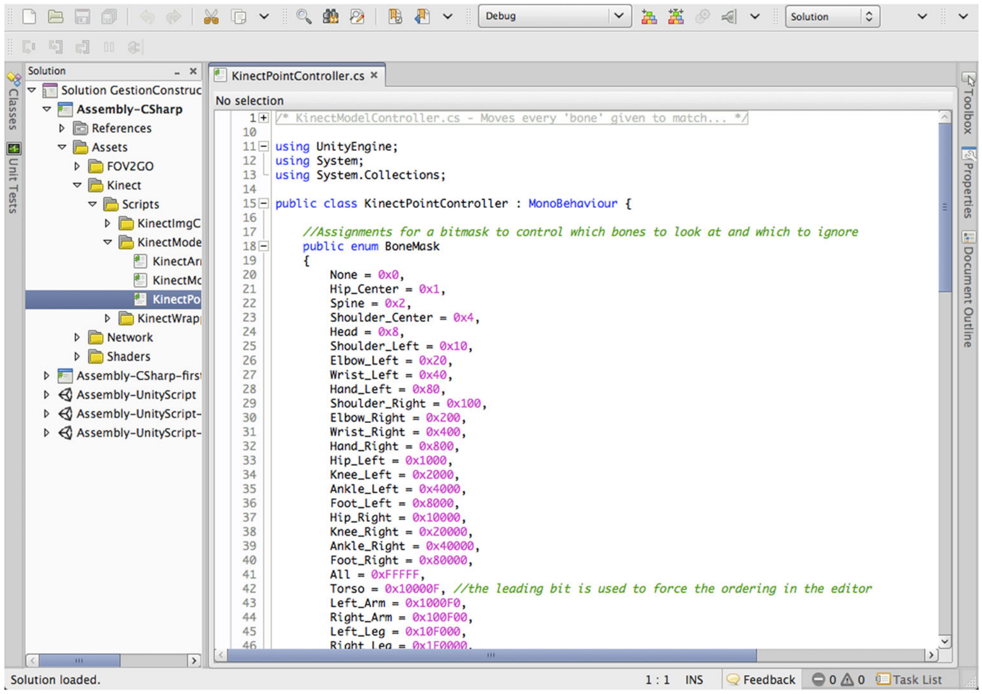


Fig. 4 Mono develop user interface

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.

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 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 Using scenario

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

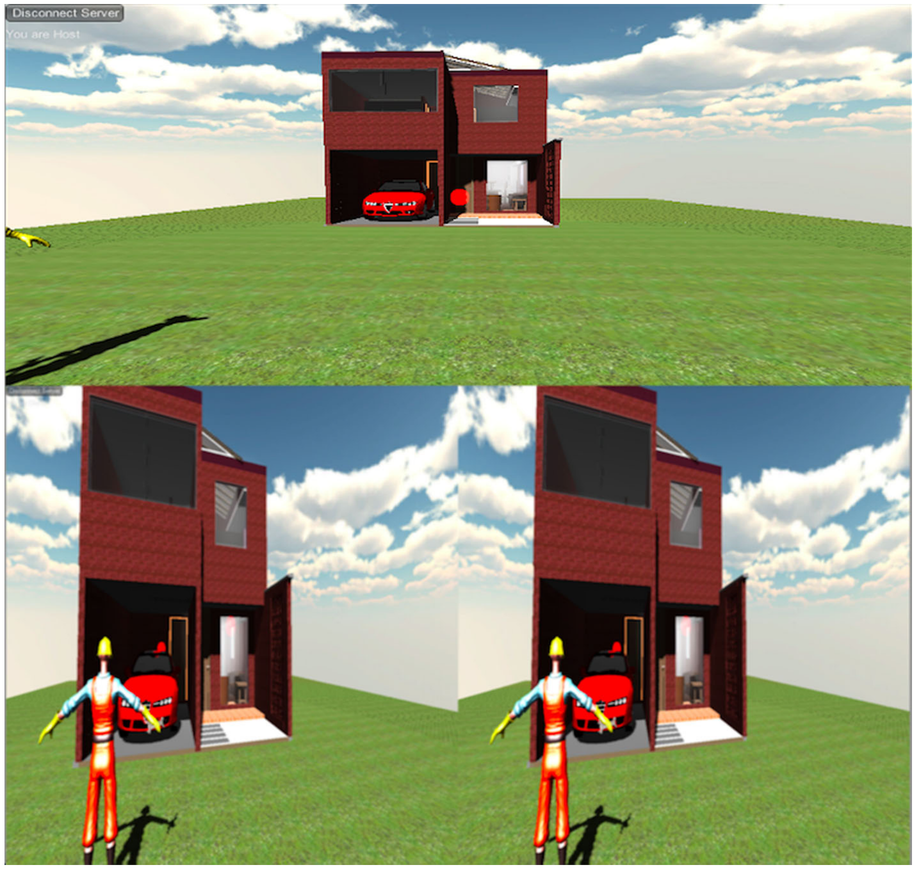


Fig. 5 *Top* Host view from a regular screen. *Bottom* Client view with stereoscopic setting on

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 5 shows an example view of the built application.

4 Experiments

Two major experiments took place during the experimentation phase of the research, named Experiment A and Experiment B. 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. Table 2 shows the variables from the experiments.

Table 3 shows the variables of the two experiments.

Table 3 Experiment variables

	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

4.1 Perception results

For the perception analysis, Technology Acceptance Model was used. 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 [6].

From the surveyed students participating in experiment A, 71 % answered that they completely agree with the state-

Table 4 Welch two sample t-test for experiment A

Welch Two Sample t-test		
Data	Model errors and plans errors	
t = 5.5549	df = 4.523	p-value = 0.003552
Alternative hypothesis:	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 5 Welch two sample t-test for experiment B

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:	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	

ment "immersive virtual reality will make it easier for me to understand a class on design review", this gives the authors a hint about the high acceptance of the tool. This also provides hints about the potential it has for performing this kind of activities. Regarding the perceived ease of use, 86 % of respondents completely agreed that interaction with the tool was easy. This is possibly related with the age range of participants.

Experiment B respondents showed that 70 % perceived the collaborative tool as a mean to perform review tasks faster. The authors believe users perceive the task being performed more quickly due to the fact that team member had its own view of the model being revised. Also, 82 % of respondents completely agree that using the tool will be useful to understand a teaching session 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 would be helpful for the task at hand, with 72 % responding that they completely agree.

4.2 Task performance results

For the performance results of the task, independent t-tests were conducted for experiment A and experiment B respectively. R statistical program was used to perform the analysis. Table 4 shows the results of *t* test from experiment A.

As *p* value is lesser than 0.05 the hypothesis that the two methods have different means can be accepted. This gives an indication that using virtual reality for a review process is a better method than reviewing 2D plans.

Table 5 shows the results for the collaborative experiment. In this case p-value is also lesser than 0.05 and we can accept the hypothesis that the two methods have different means, an indication that collaborative 3D model review is better than collaborative 2D plan review.

5 Conclusions and future work

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, levelling 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.

6 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 authors are 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.

References

1. Bourke, P.: Create side-by-side stereo pairs in the Unity game engine (2008)
2. Burdea, G., Richard, P., Coiffet, P.: Multimodal virtual reality: input output devices, system integration, and human factors. Int. J. Human Comp. Interact. **1**(8), 5–24 (1996)
3. CAMACOL: Colombia Construccion Informe 2013. Tech. rep., CAMACOL (2013)
4. Carnegie Mellon University Entertaintment Technology Center: Microsoft Kinect–Microsoft SDK—Unity3D (2014). http://wiki.etc.cmu.edu/unity3d/index.php/Microsoft_Kinect_-_Microsoft_SDK
5. CryTek: Crytek—MyCryENGINE (2014). http://mycryengine.com/index.php?conid=59
6. Davis, F.D.: Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Quarterly **13**(3), 319–340 (1989)
7. El Bibany, H.: AEC Collaborative Information Systems: From Requirements to Architecture. Automation and Robotics in Construction Xi, pp 515–522 (1994). doi:10.1016/B978-0-444-82044-0.50072-2
8. EON reality: EON Studio—Build Interactive 3D Content Quickly and Easily (2014). http://www.eonreality.com/eon-studio
9. EPIC Games Inc: Free Game Engine for Indie Game Development—UDK Unreal Developers Kit (2014). http://www.unrealengine.com/en/udk/
10. EPIC Games Inc: UDK Licensing Unreal Technology (2014). https://www.unrealengine.com/udk/licensing/purchase/
11. Fischer, X., Nadeau, J.P.: Interactive design: then and now. Res. Interac. Design **3**, 1–5 (2011)
12. Im in VR: MiddleVR for Unity (2014). http://www.imin-vr.com/middlevr-for-unity/
13. Microsoft Corporation: Getting Started with DirectX Graphics (Windows) (2014). http://msdn.microsoft.com/en-us/library/windows/desktop/hh309467(v=vs.85).aspx
14. Microsoft Corporation: Microsoft Kinect Documentation—Sensor Components and Specifications (2014). http://msdn.microsoft.com/en-us/library/jj131033.aspx
15. Microsoft Corporation: Microsoft Kinect Documentation—Skeletal Tracking (2014). http://msdn.microsoft.com/en-us/library/hh973074.aspx
16. Motion Arcade Inc: Zigfu Legacy (2014). http://zigfu.com/en/downloads/legacy/
17. NaturalPoint Inc.: OptiTrack–V100:R2—Technical specifications for the V100:R2 (2014). http://www.naturalpoint.com/optitrack/products/v100-r2/specs.html
18. Nvidia Corp: 3D Vision Pro FAQ—NVIDIA (2014). http://www.nvidia.com/object/3dvision-pro-faq.html
19. OpenGL Community: FAQ—OpenGL.org (2014). http://www.opengl.org/wiki/FAQ#What_is_OpenGL.3F
20. Ramos, M., Delgado, J., Cervantes, D., Leriche, R.: Creacion de ambientes virtuales inmersivos con software libre. Revista Digital Universitaria **8**(6) (2007)
21. Salman, A.: Building information modeling (BIM): trends, benefits, risks, and challenges for the AEC industry. Leadersh. Manage. Eng. **11**(3), 241–252 (2011)

22. Singhal, S., Zyda, M.: Networked Virtual Environments: Design and Implementation. ACM Press/Addison-Wesley Publishing Co., New York (1999)
23. Tanenbaum, A.S., Wetherall, D.J.: Introduction. In: Computer networks, 5th editio edn., p 2. Prentice Hall; 5 edn, Boston (2010)
24. The Theoretical and Computational Biophysics Group: Quad-buffered Stereo. VMD User's Guide Version: 1.9.1. Beckman Institute for Advanced Science and Technology, University of Illinois at Urbana-Champaign (2014). http://www.ks.uiuc.edu/Research/vmd/vmd-1.9.1/ug/node101.html
25. The Theoretical and Computational Biophysics Group: Side By Side and Cross Eyed Stereo. VMD User's Guide Version: 1.9.1. Beckman Institute for Advanced Science and Technology, University of Illinois at Urbana-Champaign (2014). http://www.ks.uiuc.edu/Research/vmd/vmd-1.9.1/ug/node102.html
26. Trefftz Marsic, Z.: Handling Heterogeneity in Networked Virtual Environments. Presence Teleoperators Virtual Environ **12**(1), 37–51 (2003)
27. Unity Technologies: Unity Game engine, tools and multiplatform (2014). http://unity3d.com/unity/
28. Unity Technologies: Unity High Level Networking Concepts (2014). http://docs.unity3d.com/Documentation/Components/net-HighLevelOverview.html
29. University of North Carolina at Chapel Hill: VRPN (2014). http://www.cs.unc.edu/Research/vrpn/
30. University of Southern California Institute for Creative Technologies: Flexible Action and Articulated Skeleton Toolkit (FAAST) (2014). http://projects.ict.usc.edu/mxr/faast/
31. University of Southern California Institute for Creative Technologies: FOV2GO Developer (2014). http://projects.ict.usc.edu/mxr/diy/fov2go-developer/
32. Worcester Polytechnic Institute: Unity Indie VRPN Adapter (UIVA) (2014). http://web.cs.wpi.edu/gogo/hive/UIVA/
33. WorldViz: Anaglyphic (red/cyan glases) (2014). http://docs.worldviz.com/vizard/Anaglyphic_(red_green_glases).htm
34. WorldViz: Frame sequential (2014). http://docs.worldviz.com/vizard/Frame_sequential.htm