A Project Report

on

CAD EXPLORER

Submitted for partial fulfilment of the requirements for the award of the degree
of

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING

BY

P. RISHITA REDDY
1601-14-733-137

MD.MUJTABAUDDIN FURQAN
1601-14-733-158

Under the guidance of

Smt. G. VANITHA
Asst. Professor
Department of CSE
CBIT, Hyderabad

Department of Computer Science and Engineering
Chaitanya Bharathi Institute of Technology
Hyderabad - 500075

April 2018

Chaitanya Bharathi Institute of Technology
(Regd. No. 855/2009)
(Affiliated to Osmania University, Accredited by NBA (AICTE), Accredited by NAAC (UGC), ISO Certified 9001 : 2008)
Chaitanya Bharathi P.O., CBIT Campus, Gandipet, Kokapet (V),
Rajendranagar Mandal, Ranga Reddy District, Hyderabad - 500 075
e-mail : principal@cbit.ac.in; principalcbit1979@gmail.com; Website : www.cbit.ac.in
✆ : 040 - 24193276; 24193277; 24193280; Fax : 040 - 24193278

CERTIFICATE

This is to certify that the project work entitled "CAD Explorer" is a bonafide work carried out by P. Rishita Reddy (160114733137) and Md. Mujtabauddin Furqan (160114733158) in partial fulfilment of requirements for the award of degree of Bachelor of Engineering in Computer Science and Engineering under the supervision of Smt.G.Vanitha.This has not been submitted to any other university or institute for award of degree or diploma.

Project Guide	Head of the Dept
Smt. G. Vanitha	**Dr. M Swamy Das**
Asst. Professor	Professor and Head
Department of CSE	Department of CSE
CBIT, Hyderabad	CBIT, Hyderabad

DECLARATION

We hereby declare that the project work entitled "CAD Explorer" is original and bonafide work carried out by us as a part of fulfilment for Bachelor of Engineering in Computer Science and Engineering, Chaitanya Bharathi Institute of Technology, Gandipet, Hyderabad, under the guidance of Smt.G.Vanitha, Asst. Professor, Department of CSE, CBIT.

No part of the project work is copied from books/journals/internet and wherever the partition is taken, the same has been duly referred in the text. The reported are based on the project work done entirely by us and not copied from any other source.

P. Rishita Reddy
(160114733137)

Md. Mujtabauddin Furqan
(160114733158)

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without introducing the people who made it possible and whose constant guidance and encouragement crowns all efforts with success. They have been a guiding light and source of inspiration towards the completion of the project.

We would like to express our sincere gratitude and indebtedness to our project guide, Smt.G.Vanitha who has supported us throughout our project with patience and knowledge.

We are thankful to Head of the Department, Dr. M. Swamy Das for providing excellent infrastructure and a conducive atmosphere for completing this project successfully.

We are extremely thankful to our project incharge, Smt. K. Spandana , Assistant Professor, Dept. of CSE and our project coordinator, Dr. Y. Rama Devi, Professor, Dept. of CSE, for their valuable suggestions and interest throughout the course of this project.

We are also grateful to V.Madhusudhan Rao, Associate Professor, Dept. of CSE, for letting us use his research equipment.

We convey our heartfelt thanks to the lab staff for allowing us to use the required equipment whenever needed.

Finally, we would like to take this opportunity to thank our families for their support through the work. We sincerely acknowledge and thank all those who gave directly or indirectly their support in completion of this work.

P. Rishita Reddy
(160114733137)

Md. Mujtabauddin Furqan
(160114733158)

ABSTRACT

The goal of Augmented Reality (AR) is to improve and enhance our perception of the surroundings by combining sensing, computing and display technologies. Augmented Reality is a variation of Virtual Reality. Augmented Reality allows the user to see the virtual objects superimposed upon with the real world. This can be done with the help of a Head Mounted Display. In this project, we would be using Microsoft Hololens. The basic aim of our project is to develop an application for Hololens. This application would be designed and developed in a 2D environment and then 3D elements can be added to it, with the help of Unity 3D. This project will give us good knowledge on Augmented Reality and hence gives us an opportunity to explore a path not taken.

The project we are going to work on aims at displaying 3D House Modelling files and adding functionality such as zooming and modifying the models. The 3D House Modelling files could be from various Engineering disciplines. The aim is to facilitate viewing of these models in a 3D space.

List of Figures

Table of Contents

1. Introduction

Microsoft HoloLens is a pair of mixed reality smart glasses developed and manufactured by Microsoft. HoloLens gained popularity for being one of the first computers running the Windows Mixed Reality platform under the Windows 10 operating system. The HoloLens can trace its lineage to Kinect, an add-on for Microsoft's Xbox gaming console that was introduced in 2010.

The HoloLens is a head-mounted display unit connected to an adjustable, cushioned inner headband, which can tilt HoloLens up and down, as well as forward and backward. To wear the unit, the user fits the HoloLens on their head, using an adjustment wheel at the back of the headband to secure it around the crown, supporting and distributing the weight of the unit equally for comfort, before tilting the visor towards the front of the eyes.

Through the use of the HPU, the unit uses sensual and natural interface commands—gaze, gesture, and voice—sometimes referred to as "GGV", inputs. Gaze commands, such as head-tracking, allows the user to bring application focus to whatever the user is perceiving. "Elements"—or any virtual application or button—are selected using an air tap method, similar to clicking an imaginary computer mouse. The tap can be held for a drag simulation to move an element, as well as voice commands for certain commands and actions.

The project we are going to work on aims at displaying 3D House Modelling files and adding functionality such as zooming and modifying the models. The 3D House Modelling files could be from various Engineering disciplines. The aim is to facilitate viewing of these models in a 3D space.

1.1 A Crash course in Realities

Trying to explain nascent technology has its challenges; attempting to understand a groundbreaking product like the HoloLens and how it differs from other virtual reality devices, is something else entirely. Microsoft HoloLens is a mixed reality device where the user experiences both the computer-generated and the natural world as a hybrid construct. When wearing the headset, virtual objects are placed alongside actual physical objects to create a unified environment of physical and digitally created experiences. This mixed reality experience creates an experiential relationship as the lines between real and virtual begin to blur. In contrast to the mixed reality environment, there are completely immersive virtual reality environments. The viewer wears goggles and visually enters a digitally manufactured world and is able to virtually move around within the fabricated world. Devices such as Samsung's Oculus Rift and HTC Vive support these types of immersive experiences.

Augmented reality is another type of virtual experience that offers information and insight into the environment. These experiences access information from the web as the technology, i.e. mobile phone, layers the information on top of the user's visual field while not interfering with the experience of the environment. Pokémon Go is a simple example of displaying data in the environment while the user is still fully engaged in the real world.

In the case of HoloLens, the novel interface has digitally produced, full-color virtual objects — holograms — that appear on lenses in the headset right in front of the viewer's eyes. The user's brain responds to the combined virtual/actual environment as a unified experience. Think Princess Leia in the original Star Wars. The challenge was to create viable proposals while the HoloLens product itself was being developed. The technology and creative parameters were in a constant state of evolution as new development was being made on the engineering and software teams on a daily basis. There were over 25 original proposals from students across a variety of creative disciplines including design, art, theater, and dance. The projects selected were based on several factors beginning with the quality of their overall concept coupled with the feasibility of executing the idea in a two-month timeframe and within the technical capabilities of the current iteration of the HoloLens device.

Following are the main concepts which are used in the application.

1.Gaze

2. Gesture

3. Spatial Mapping

1.2 Gaze

Gaze is a first form of input and is a primary form of targeting within mixed reality. Gaze tells where the user is looking in the world and lets us determine their intent. On HoloLens, interactions should generally derive their targeting from the user's gaze, rather than trying to render or interact at the hand's location directly. Once an interaction has started, relative motions of the hand may be used to control the gesture, as with the manipulation or navigation gesture. With immersive headsets, you can target using either gaze or pointing-capable motion controllers.

1.2.1 Uses of Gaze

▪ An app can intersect gaze with the holograms in your scene to determine where the user's attention is.

▪ An app can target gestures and controller presses based on the user's gaze, letting the user select, activate, grab, scroll, or otherwise interact with their holograms.

▪ An app can let the user place holograms on real-world surfaces, by intersecting their gaze ray with the spatial mapping mesh.

▪ An app can know when the user is not looking in the direction of an important object, which can lead your app to give visual and audio cues to turn towards that object.

1.3 Gestures

Gestures allow users take action in mixed reality with their hands. For HoloLens, gesture input lets you interact with your holograms naturally. To take actions, gestures use gaze as targeting mechanism. The combination of gaze and a select gesture results in a gaze-and-commit interact.

Below mentioned are the different gestures used in the app along with the way to apply them.

▪ Tap - Tap gestures (as well as the other gestures below) react only to select presses. The Tap gesture may be used to perform actions such as selecting Holograms.

▪ Hold - Hold gestures are similar to a touch tap-and-hold and can be used to take a secondary action, such as picking up an object instead of activating it, or showing a context menu.

▪ Manipulation - Manipulation gestures can be used to move, resize or rotate a hologram when you want the hologram to react 1:1 to the user's hand movements. One use for such 1:1 movements is to let the user draw or paint in the world. The initial targeting for a manipulation gesture should be done by gaze or pointing. Once the press starts, any manipulation of the object is then handled by hand movements, freeing the user to look around while they manipulate.

1.4 Spatial Mapping

Spatial mapping provides a detailed representation of real-world surfaces in the environment around the HoloLens, allowing developers to create a convincing mixed reality experience. By merging the real world with the virtual world, an application can make holograms seem real. The two primary object types used for spatial mapping are the 'Spatial Surface Observer' and the 'Spatial Surface'.

The application provides the Spatial Surface Observer with one or more bounding volumes, to define the regions of space in which the application wishes to receive spatial mapping data. For each of these volumes, spatial mapping will provide the application with a set of Spatial Surfaces.These volumes may be stationary (in a fixed location with respect to the real world) or they may be attached to the HoloLens (they move, but do not rotate, with the HoloLens as it moves through the environment). Each spatial surface describes real-world surfaces in a small volume of space, represented as a triangle mesh attached to a world-locked spatial coordinate system.

1.5 Problem Definition

The main idea behind this project it to be able to view 2D models in 3D using Augmented Reality technology offered by the Microsoft HoloLens. The goal of Augmented Reality (AR) is to improve and enhance our perception of the surroundings by combining sensing, computing and display technologies. Augmented Reality is a variation of Virtual Reality. Augmented Reality allows the user to see the virtual objects superimposed upon with the real world. This can be done with the help of Head Mounted Display.The basic aim of our project is to develop an application for Hololens.

This application would be designed and developed in a 2D environment and then 3D elements can be added to it, with the help of Unity 3D. The project we are going to work on aims at displaying AutoCAD files and adding functionality such as zooming and modifying the files. The AutoCAD files could be from various Engineering disciplines. In our project, we will be displaying it using a model of a house and allowing modifications such as changing the dimensions of a room by moving the walls around.

Initially, the user will be able to view the objects in 3D. He can then increase/decrease the size of the room by certain gestures which will be recognized by the Hololens. If Augmented Reality is not used, the files are viewed in 2D on a computer which will not give the user proper understanding of the dimensions. In this project, the 3D model will show the dimensions in a 1:1 ratio for better understanding.

1.6 Organization of Report

This thesis is organized as follows:

- Section 1 deals with Introduction to the project, gaze, gestures, spatial mapping and problem description.
- Section 2 explains about Literature Survey conducted before starting off the project.
- Section 3 gives the design details of the system like algorithm used for the system and design(UML) diagrams. It also explains the implementation of the system and talks about system requirements.
- Section 4 gives details about result analysis.
- Section 5 discusses the conclusion as well as future works.
- These sections are followed by References.
- The last section describes the Appendix.

2. Literature Survey

2.1 Introduction

Virtual Reality or Virtual Environments is a computer simulated environment that gives the user the experience of being present in that environment. It is a 3-Dimensional computer generated environment. VR provides the effects of a concrete existence without having a concrete existence. VR not only provides immersions of vision but also of sound and tactile feedback. Basically, VR is a theory based on the human desire to escape the real world boundaries and this is done by embracing the cyber world. It is a new form of human machine interaction that is beyond keyboard, mouse or even touch screen for that matter.

Augmented Reality is a variation of Virtual Reality or Virtual Environments. AR allows the user to see the real world, with virtual objects superimposed upon or composited with the real world. Therefore, AR supplements reality, rather than completely replacing it.

Fig 2.1 Person using Microsoft HoloLens

Fig 2.1 shows a person using the Hololens and also displays what the user is viewing. We can see the user trying to interact with the bike hologram with her finger.

Microsoft HoloLens goes beyond augmented reality and virtual reality by enabling the user to interact with three-dimensional holograms blended with the world. Microsoft HoloLens is more than a simple heads-up display, and its transparency means you never lose sight of the world around you. High-definition holograms integrated with your real world will unlock all- new ways to create, communicate, work, and play. Microsoft HoloLens intelligently maps the room you're in, blending holograms with the environment around you. Pin holograms to physical locations you choose so that your room becomes the canvas for your holographic projects and games. With Microsoft HoloLens, you can interact with holograms and everyday objects together. It puts you at the center of a world that blends holograms with reality.

Fig 2.2 Microsoft HoloLens

Fig 2.2 depicts how the Microsoft Hololens looks. It shows the appearance of the device, containing lenses and a headband.

2.2 Description

Microsoft HoloLens is an augmented reality computing platform, features see-through, holographic, high-definition lenses and spatial sound so that user can see and hear holograms in the world around you. Which is developed by Microsoft that is set to be introduced in the Windows 10 operating system. It is completely untethered–no wires, phones, or connection to a PC is needed. Microsoft HoloLens allows you to pin holograms in your physical environment and provides a new way to see your world. With Microsoft Holographic APIs, augmented reality features can be readily implemented on a wide range of Windows 10 devices.

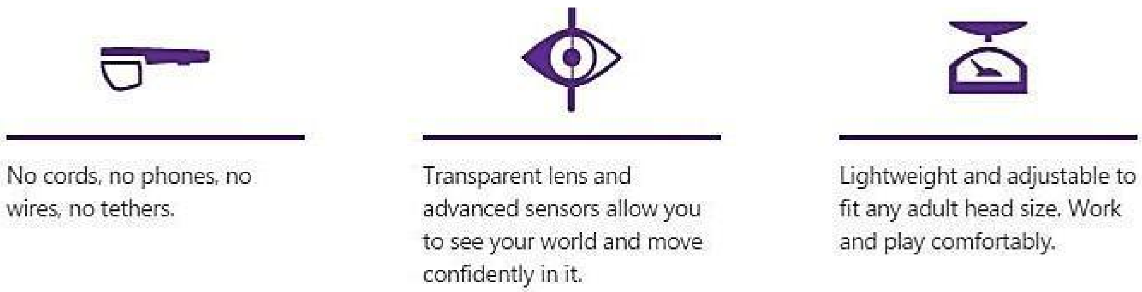

Fig 2.3 HoloLens description

Fig 2.3 describes the Hololens. It explains how the Hololens doesn’t require any wires, cords and is hence, untethered. It has a transparent lens as well as advanced sensors which help the user move about the world confidently. The hololens is also lightweight and adjustable to fit any head size as required.

Microsoft HoloLens is the first holographic computer running Windows 10. Microsoft HoloLens equipped with advanced sensors and a new Holographic Processing Unit (HPU) that understands the world around the user, and is able to run without any wires while processing terabytes of data from the sensors in real-time. Holograms are made entirely of light. Holographic objects can be viewed from different angles and distances, just like physical objects, but they do not offer any physical resistance when touched or pushed because they don’t have any mass. Holograms can be two-dimensional, like a piece of paper or a TV screen, or they can be three-dimensional, just like other physical objects in your real world. The holograms you’ll see can appear life-like, and can move, be shaped, and change according to interaction with users or the physical environment in which they are visible.

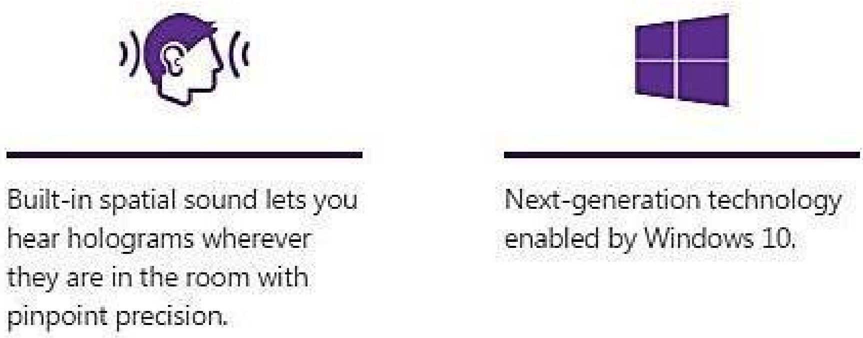

Fig 2.4 Description of Microsoft Hololens

Fig 2.4 talks about the feature of Spatial sound which is built into the hololens, and lets the user hear holograms with great precision. It has next gen technology enabled by Windows 10.

2.3 Working

The idea behind augmented reality is nothing new and early prototypes date back to the late 60s. Early achievements include heads up displays (HUDs) in jet fighters and helicopters, help with targeting and provide night vision. Such advancements were then passed on to civil aviation and later cars. The technology behind augmented reality systems involves numerous fields of research including signal processing and tracking systems, graphics, user interfaces, human factors, wearable and mobile computing, networking or information visualization. The requirements for an AR system depend greatly on its intended application. A visual AR system revolves around a display, a precise tracking system which provides accurate environmental information to keep virtual elements synchronized with the real ones and extensive computational power to handle the real-time requirements of augmented reality. AR systems must satisfy at least three conditions in order to function properly.

•Trackers must be accurate to a small fraction of a degree in orientation and a few millimetres in position.

•The combined latency of the tracker and the graphics engine must be very low.

•The tracker must work at long ranges.

Microsoft HoloLens is equipped with a Holographic processing unit which is a specialized electronic circuit designed as a co -processor chip (along with CPU and GPU) dedicated to process and seamlessly merge real environments with virtually generated data and graphics. Essentially, HPU's try to understand the world around itself. The HPU was exclusively developed for Microsoft's HoloLens. According to Microsoft the HPU is capable of, processing terabytes of information from all its sensors in real time. The HPU is integrated into the Microsoft HoloLens, which facilitates the merging of virtual objects with the user's environment in real time. The device has transparent lenses, which is said to be of three layers.

These helps the user to view the holographic object in its depth and reality. The device auto adjusts to the users eye pitch and then creates the objects accordingly. Microsoft's HoloLens is not actually producing 3D images that everyone can see. Instead of everyone walking into a room made to reproduce 3D images, HoloLens show images only the wearer can see. HoloLens is not trying to transport you to a different world, but rather bring the wonders of a computer directly to the one you're living in. it’s just overlaying images and objects onto our living rooms.

Fig 2.5 Sensors & Cameras on Microsoft HoloLens

Fig 2.5 shows the sensors and the camera present on the Hololens. It specifies their location and provides us with a closer look.

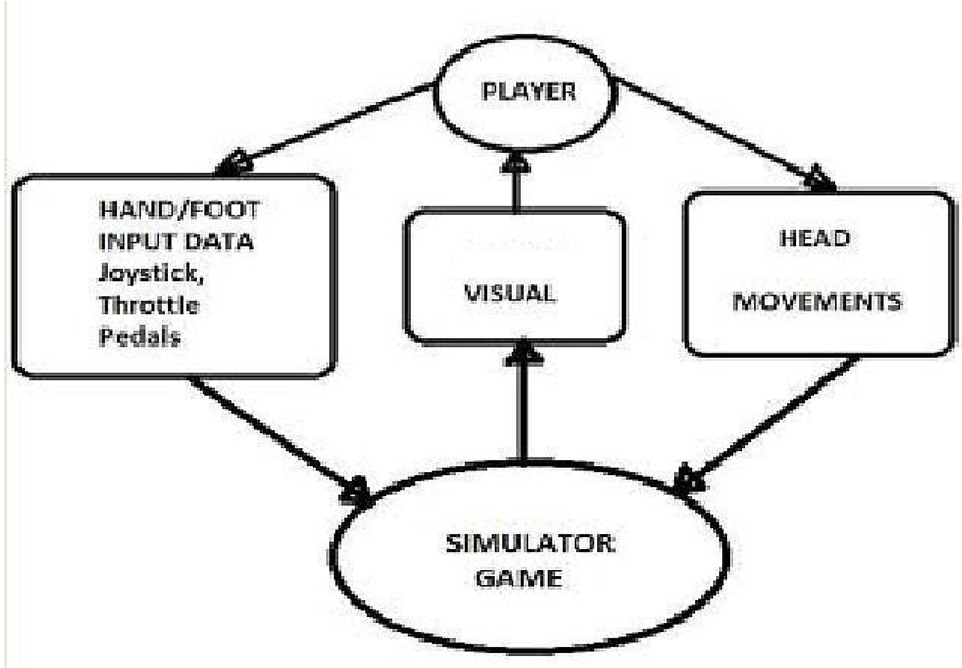

Fig 2.6 Flowchart for Simulator game

Fig 2.6 shows the flowchart for a simulator game. The player has three ways of interacting with the game, namely- hand or foot input data (using joystick), visual interaction as well as head movements. Whatever is done by the user is simulated in the game.

The device has camera that looks at the room, so the HoloLens knows where tables, chairs and other objects are. It then uses that information to project 3D images on top of and even inside them. The HPU processes the data from these cameras to identify the objects in the room, their shapes, and positions. The device has a plethora of sensors to sense your movements in a room and it uses this information along with layers of colored glass to create images you can interact with or investigate from different angles. The devices uses different kinds of sensors to take inputs from the user and to understand the surrounding. Its uses motion tracker that use a gyroscope, accelerometer, and magnetometer to sense user's head motion. Sensors are used to read the light intensity, Motion sensors to detect the user's actions, IR sensors to detect the obstacles, a depth camera, GPS to locate the user, orientation trackers and so on. Microsoft HoloLens captures even slight movement of the user and acts accordingly to it.

2.4 Applications

The possibilities enabled by the Microsoft HoloLens see a wide array of use-cases developers, commercial organizations, designers, creators, and those seeking a whole new way to be entertained will find unique value in Microsoft HoloLens. For developers, Windows 10 and Microsoft HoloLens are the premier holographic platform that enables him to unleash the creativity and be among the first to shape holographic computing. Microsoft HoloLens will be a revolutionary tool for businesses. It will transform how companies, designers, and creators work with three-dimensional data to bring products and information to life. Microsoft had given demo on four application using the prototype. They are Building a 3D Model, Installing a Light Switch with a help from a technician over a Skype call, exploring the surface of Mars, Minecraft-Like Gaming.

Fig 2.7 Person using HoloLens to build a model

Fig 2.7 shows a person using the Hololens to build a model of the rocket. It can be understood, as he initially drew the sketch on a piece of paper and is then implementing it.

Microsoft HoloLens can be used to design 3D Models in real-time using HoloStudio, a 3D modelling tool. You can walked around the hologram, grab tools from a holographic control panel, and then used a combination of voice and gestures to build and shape model. These models can then be sent to a 3D printer for manufacturing.

Fig 2.8 Skype enabled HoloLens version

Fig 2.8 shows a Skype enabled version implemented in Hololens, through which two users can speak using holograms.

A HoloLens-enabled version of Skype can be used to interact between two personals and can remotely assist the other to do some tasks. A small window appears in virtual field so the user could video chat. But using the HoloLens, the person at the other end could see what user is watching. The person can assist the user by drawing an arrow or annotations that shows exactly what actions to perform.

Fig 2.9 Exploring mars surface using HoloLens

Fig 2.9 shows a user exploring the surface of mars by using Hololens. It makes the user feel as though he is moving though that simulated environment.

Microsoft has been working with NASA's Jet Propulsion Laboratory (JPL) to create a holographic version of Mars for research and maintenance purposes. Using the HoloLens the user can feel the experience of standing near the Mars rover, surrounded by the Red Planet's vistas. Looking down he can see the rocks of the martian land just inches away. This entire environment had to be constructed from photos taken from the mars rover.

Fig 2.10 Gaming using HoloLens

Fig 2.10 shows the user gaming by using Hololens. This makes the user feel as though he is in a different environment, and makes gaming more fun.

Playing games using the HoloLens will be a different level of experience. Microsoft had given demos using their own block building game which can turn the small living room to be filled with blocky castles, on the coffee table and along the wall. User could walk around the structures, gaze upon individual blocks, and then make changes to them using the air click. Voice command lets us change the tools quickly. We can drill holes in the castle and look down through the virtual floor into the levels below. It was immersive, but probably a little slower than it might be with a mouse and keyboard.

3. Methodology

Methodology includes the background work that has been done before starting the implementation of the application. In this section we talk about the system design, the implementation details as well as the System Requirements for the application.

3.1 System Design

It includes the the proposed algorithm, the diagrammatic representations by using UML diagrams. The different UML diagrams include DFD, Activity Diagram, State Chart Diagram, Sequence Diagram and Use Case Diagram.

3.1.1 Proposed Workflow

The basic workflow of the application is as follows

CAD Explorer:

- Event: Application opened
  - Load the environment
  - Spatial mapping of 3D objects
- Event: User uses airtap gesture
  - Hologram is selected and menu is displayed
- Event: User selects Remove button
  - The hologram is removed
- Event: User selects Modify button
  - SubEvent: User drags edge
    - The hologram gets scaled
  - SubEvent: User taps, holds and moves
    - The hologram moves

3.1.2 Diagrammatic Representation

3.1.2.1 Data Flow Diagram

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated.

3.1.2.1.2 Level 0

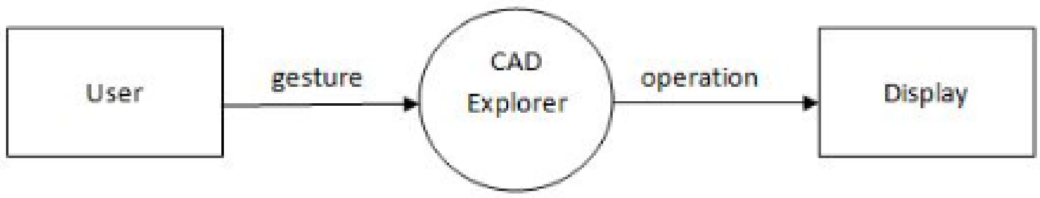

Fig 3.1 Level 0 DataFlow Diagram

Fig 3.1 shows a Level 0 DFD, which consists of external entities, namely- User and Display and a process called CAD Explorer. User performs a gesture, which will be given as input to the application CAD Explorer. The output given by the application will be the operation based on the gesture, which is then sent to Display.

3.1.2.1.2 Level 1

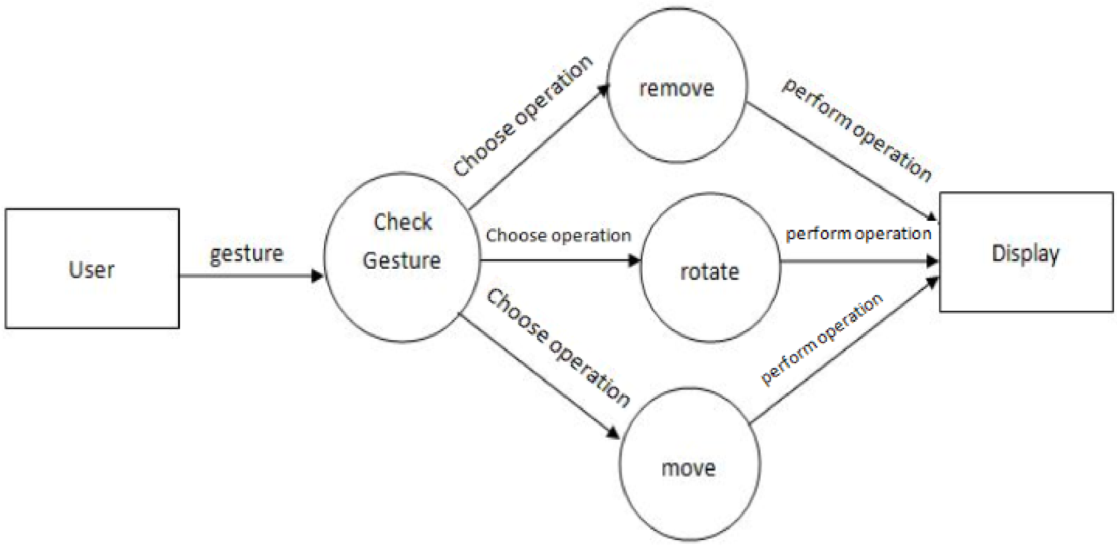

Fig 3.2 Level 1 DataFlow Diagram

Fig 3.2 shows a Level 1 DFD, which consists of external entities, namely – User and Display. It also consists of processes Check Gesture, Remove, Rotate and Move. The user initially performs a gesture, which is sent as input to the process Check gesture, which checks among the processes remove, rotate and move. Depending on the gesture selected, the operation is performed and sent to the entity Display which shows it on the Hololens.

3.1.2.2 Activity Diagram

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another.

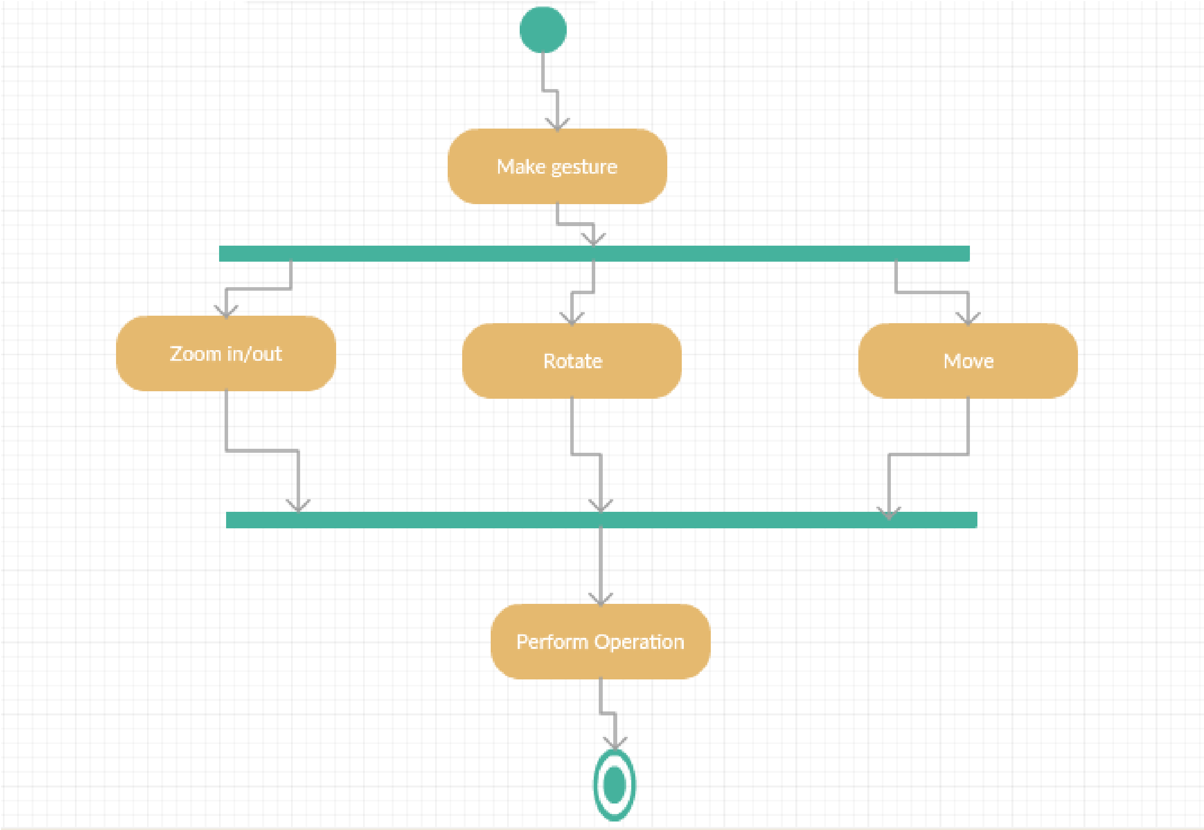

Fig 3.3 Activity Diagram

Fig 3.3 shows activity diagram in which, the user initially needs to make a gesture, after which he has to do one of three operations, namely – Rotate, Move or Remove the hologram. Whatever he selects will lead to performing the operation, and the activities are completed.

3.1.2.3 Statechart Diagram

A Statechart diagram describes a state machine. State machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events.

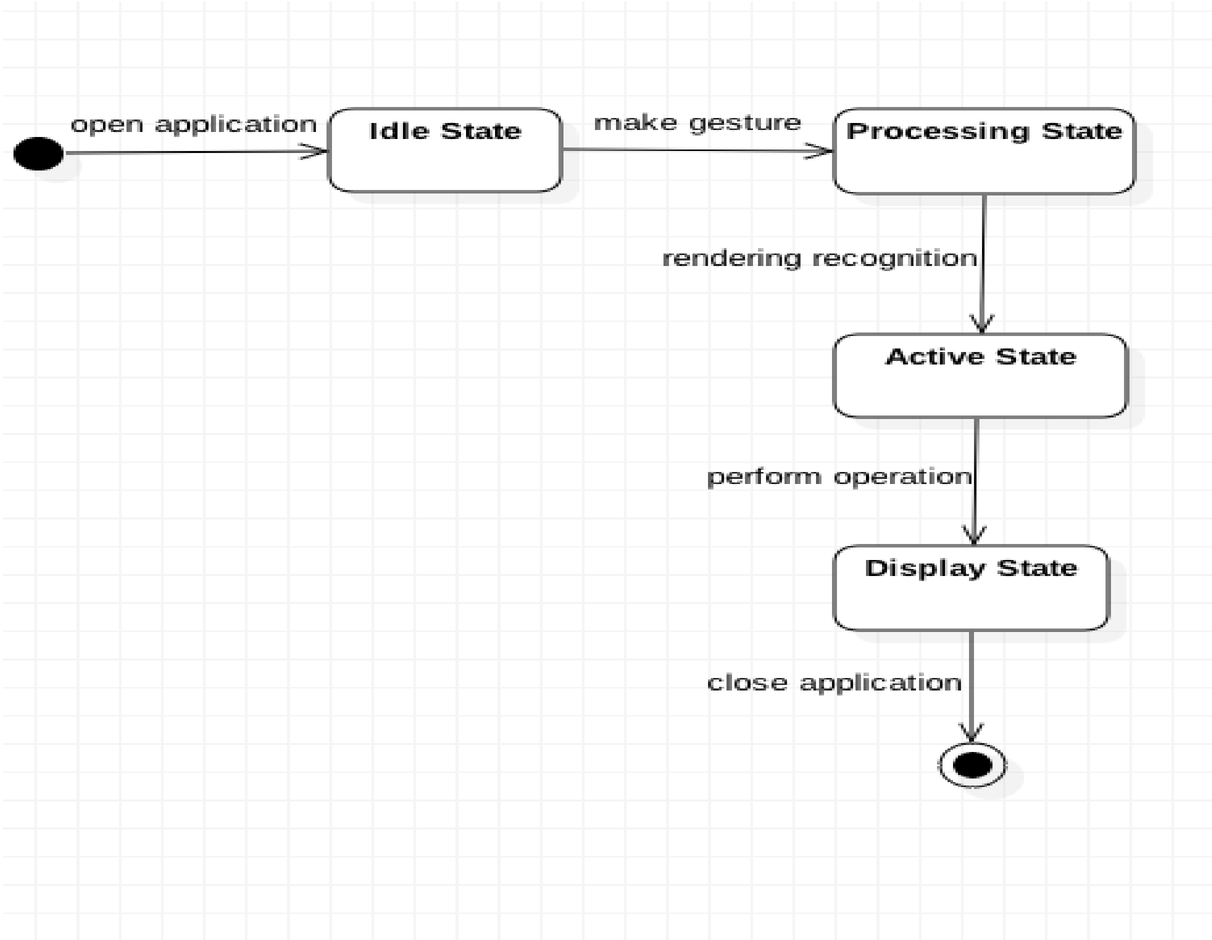

Fig 3.4 StateChart Diagram

Fig 3.4 shows the statechart diagram in which initially, when the application is opened, it goes into Idle state, after which the user makes a gesture and goes into the processing state. Rendering recognition is done which takes the system into active state. Upon performing the operation, it goes into display state. As the application is closed, the diagram goes into final state.

3.1.2.4 Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.

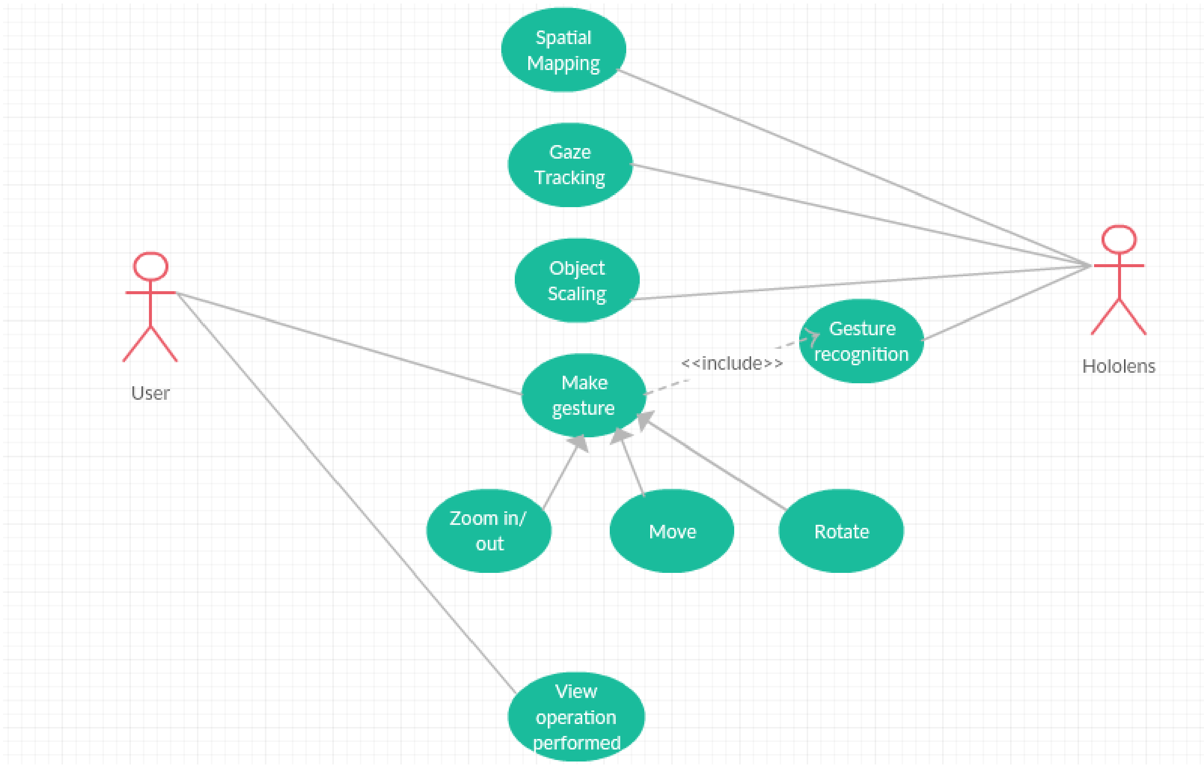

Fig 3.5 UseCase Diagram

Fig 3.5 contains two actors- the user and the Hololens. It has different use cases of Gaze tracking, Object scaling which are done by Hololens. The user can make a gesture, namely- move, rotate or remove which is recognized by the Hololens. The user can then view the operation performed.

3.1.2.5 Sequence Diagram

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

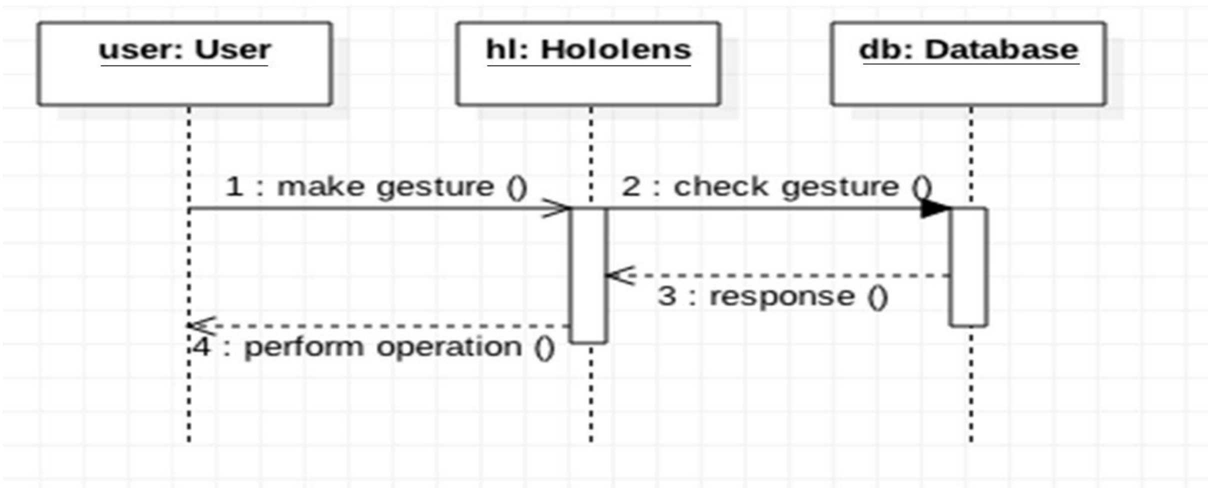

Fig 3.6 Sequence Diagram

Fig 3.6 has three lifelines- the user, hololens and the database. Initially, the user makes a gesture to the hololens, after which the hololens checks the gesture in the database. The database then responds to the hololens about the validity of the gesture. The hololens then performs the appropriate operation which can be viewed by the user.

3.2 Implementation of Proposed workflow

Implementing the Project on a Game Engine is the first Module that is to be completed. In Unity 5.5, the entire Scene, including all its menu and User Interface is designed. GameObjects constitute the fundamental units of the Game Engine.

Primarily, Unity works with Scenes. A Scene is, in the most basic terms, a representation of the state of the system. In a Scene, multiple GameObjects may be placed, and their components changed. This is the fundamental idea behind designing a app in Hololens.

Unity provides a user with a Camera and a Directional Light, both of which are crucial to ensuring that all elements in a Project are captured. Any user can only view objects from the standpoint of the Camera.

This application would be designed and developed in a 2D environment and then 3D elements can be added to it, with the help of Unity 3D. The project we are going to work on aims at displaying AutoCAD files and adding functionality such as zooming and modifying the files. The AutoCAD files could be from various Engineering disciplines. In our project, we will be displaying it using a model of a house and allowing modifications such as changing the dimensions, rotating as well as moving objects in space.

Initially, the user will be able to view the objects in 3D. He can then increase/decrease the size of the room by certain gestures which will be recognized by the Hololens. If Augmented Reality is not used, the files are viewed in 2D on a computer which will not give the user proper understanding of the dimensions. In this project, the 3D model will show the dimensions in a 1:1 ratio for better understanding.

3.2.1 Exporting and Debugging on Visual Studio

Microsoft Visual Studio is an IDE that houses the code required to run an Application. Visual Studio debugs and compiles the code that is responsible for the App to run.

It must be understood that Unity only helps achieve the design of the app i.e the outer shell. The code required to ensure that the holograms as well as the gestures applied in the app perform their functions, and other necessary actions must be placed in Visual Studio, and then transferred to Unity.

Following are a few code snippets which help us understand the different functionalities implemented.

```
case "BoxCollider":
    BoxCollider bc = colliders[i] as BoxCollider;
    Bounds boxBounds = new Bounds(bc.center, bc.size);
    boxBounds.GetCornerPositions(bc.transform, ref corners);
    boundsPoints.AddRange(corners);
    break;

case "MeshCollider":
    MeshCollider mc = colliders[i] as MeshCollider;
    Bounds meshBounds = mc.sharedMesh.bounds;
    meshBounds.GetCornerPositions(mc.transform, ref corners);
    boundsPoints.AddRange(corners);
    break;
```

The main part of the application is the implementation of a bounding box, which appears whenever a hologram is selected. To implement a bounding box, Colliders need to be implemented which help in assessing the hologram which has been selected. The code snippet for a Box Collider is shown above.

If the object with the Collider needs to be moved during gameplay then you should also attach a Rigid body component to the object. The Rigid body can be set to be kinematic if you don't want the object to have physical interaction with other objects.

The code which selects the Active Bounding Box at any given time is as follows

```
// Try to find our bounding box
if (boundingBox == null)
{
    boundingBox = ManipulationManager.Instance.ActiveBoundingBox;
}

// Try to find our toolbar
if (toolbar == null)
{
    toolbar = ManipulationManager.Instance.ActiveAppBar;
}
```

The code snippet above is a part of the BoundingBox Manipute.cs which specifies the code that helps us in modifying our holograms when they are inside the Bounding Box. This particular code activates the bounding box and the app bar when they have been activated with the help of Air tap Gesture and cursor.

When a hologram is selected, an app bar appears using which modifications on the hologram can be performed. The menu that will be displayed has the following code.

The App bar is an object-level menu containing a series of buttons that displays on the bottom edge of a hologram's bounds. This pattern is commonly used to give users the ability to remove and adjust holograms.

The bounding box is a state for objects editable in a user’s environment. It provides the user an affordance that the object is currently adjustable. The corners tell the user that the object can also scale. This visual affordance shows users the total area of the object – even if it’s not visible outside of an adjustment mode. This is especially important because if it weren’t there, an object snapped to another object or surface may appear to behave as if there was space around it that shouldn’t be there.

```
switch (obj.name) {
    case "Remove":
        // Destroy the target object
        GameObject.Destroy(boundingBox.Target);
        // Set our bounding box to null so we'll disappear
        boundingBox = null;
        break;

    case "Adjust":
        // Make the bounding box active so users can manipulate it
        State = AppBarStateEnum.Manipulation;
        break;

    case "Hide":
        // Make the bounding box inactive and invisible
        State = AppBarStateEnum.Hidden;
        break;

    case "Show":
        State = AppBarStateEnum.Default;
        break;

    case "Done":
        State = AppBarStateEnum.Default;
        break;

    default:
        break;
}
```

The above code snippet specifies the different predefined operations that an App Bar can perform. The App bar is a object-level menu containing a series of buttons that displays on the bottom edge of a hologram's bounds. This pattern is commonly used to give users the ability to remove and adjust holograms. The different operations that it can perform are Remove the hologram altogether from the scene thus with respect to this application we can remove any wall or any furniture that we would no longer desire in that specific scene.

The adjust operation lets us modify any hololens in any way we want to. It mainly lets us perform three operation Drag, Rotate and Scale. Under the Rotate operation we we can perform RotateX, RotateY, RotateZ thus letting us rotate the hologram in any direction possible. The Scale Operation has four different predefined operations namely Scale Uniform, ScaleX, ScaleY, ScaleZ thus letting us expand or constrict the hologram in any direction possible. it also lets us modify the shapes of the holograms as well. Scale Uniform lets us uniformly expand or compress the hologram in all the dimensions.

The Drag operation lets us move the hologram to any position we desire thus letting us modify the walls, furniture or any other part of our house to any part of the scene, thus making each object truly mobile.

The ability to rotate the hologram is performed by the implementation shown below.

```
case OperationEnum.RotateX:
    // transformHelper.Rotate(-smoothVelocity.y * RotateMultiplier * 360, 0f, 0f, Space.World);
    transformHelper.RotateAround(transformHelper.position, transform.right, smoothVelocity.magnitude * Vector3.Dot(proj.normalized, orthogonalVect.normalized) * -360);
    break;

case OperationEnum.RotateY:
    // transformHelper.Rotate(0f, smoothVelocity.x * RotateMultiplier * 360, 0f, Space.World);
    transformHelper.RotateAround(transformHelper.position, transform.up, smoothVelocity.magnitude * Vector3.Dot(proj.normalized, orthogonalVect.normalized) * -360);
    break;

case OperationEnum.RotateZ:
    // transformHelper.Rotate(0f, 0f, smoothVelocity.x * RotateMultiplier * 360, Space.World);
    transformHelper.RotateAround(transformHelper.position, transform.forward, smoothVelocity.magnitude * Vector3.Dot(proj.normalized, orthogonalVect.normalized) * -360);
    break;
```

In this snippet it shows how each type of rotate is made possible. Under the Rotate operation we we can perform RotateX, RotateY, RotateZ thus letting us rotate the hologram in any direction possible. RotateX lets us rotate the hologram along the x-axis. The RotateY lets us rotate the hologram along the y-axis. The RotateY lets us rotate the hologram along the z-axis.

The App bar is a object-level menu containing a series of buttons that displays on the bottom edge of a hologram's bounds. This pattern is commonly used to give users the ability to remove and adjust holograms.

The bounding box is a state for objects editable in a user’s environment. It provides the user an affordance that the object is currently adjustable. The corners tell the user that the object can also scale. This visual affordance shows users the total area of the object – even if it’s not visible outside of an adjustment mode. This is especially important because if it weren’t there, an object snapped to another object or surface may appear to behave as if there was space around it that shouldn’t be there.

The code written below shows how the hologram may be scaled or resized.

```
case OperationEnum.ScaleUniform:
case OperationEnum.ScaleX:
case OperationEnum.ScaleY:
case OperationEnum.ScaleZ:
    // Translate velocity direction based on camera
    Vector3 orientedVelocity = Camera.main.transform.TransformDirection(smoothVelocity);
    // See whether handle is to left or right of gizmo center
    Vector3 handleScreenPoint = Camera.main.WorldToScreenPoint(activeHandle.transform.position);
    Vector3 gizmoScreenPoint = Camera.main.WorldToScreenPoint(targetBoundsWorldCenter);
    float dragAmount = orientedVelocity.x;
    if (handleScreenPoint.x > gizmoScreenPoint.x)
    {
        dragAmount = -dragAmount;
    }
    transformHelper.localScale += Vector3.one * (dragAmount * ScaleMultiplier);
    break;
```

In this snippet it shows how each type of rotate is made possible. Under the Scale operation we we can perform ScaleX, ScaleY, ScaleZ , Scale Uniform thus letting us Scale the hologram in any direction possible. ScaleX lets us expand or compress the hologram along the x-axis. The ScaleY lets us scale the hologram along the y-axis. The ScaleY lets us expand or compress the hologram along the z-axis. Scale Uniform lets us uniformly expand or compress the hologram along along all three dimensions.

The code snippet for moving the selected hologram is as follows.

```
case OperationEnum.Drag:
    transformHelper.position -= (smoothVelocity * DragMultiplier);
    break;
```

The code snippet above shows us the part of code that lets us move the hologram. The Drag operation lets us move the hologram to any position we desire thus letting us modify the walls, furniture or any other part of our house to any part of the scene, thus making each object truly mobile.

3.2.2 Debugging the App in Visual Studio

After the scripts are written in Visual Studio, they need to be imported into the hierarchy and to the inspector panel of Unity according to the functionality of the script. For example- The furniture items need to be selected and placed in our application, therefore Gaze and Gesture functionality scripts needs to be imported in the inspector panel of the furniture items.

Later, we need to save the scene and go to **File→ Build Settings** in Unity. Now the following steps needs to be performed.

- Change the platform to **Windows Store.**
- Change the SDK from **Windows 8.1 to Universal 10.**
- Change the UWP Build Type to **D3D.**
- **Check** the Unity C# Projects.
- **Enable** the **Spatial Perception** and **Microphone** in the player settings.
- Select on the **Player Settings** button and from the right-side inspector section, select "**Virtual Reality Supported**" from **Other Settings**.

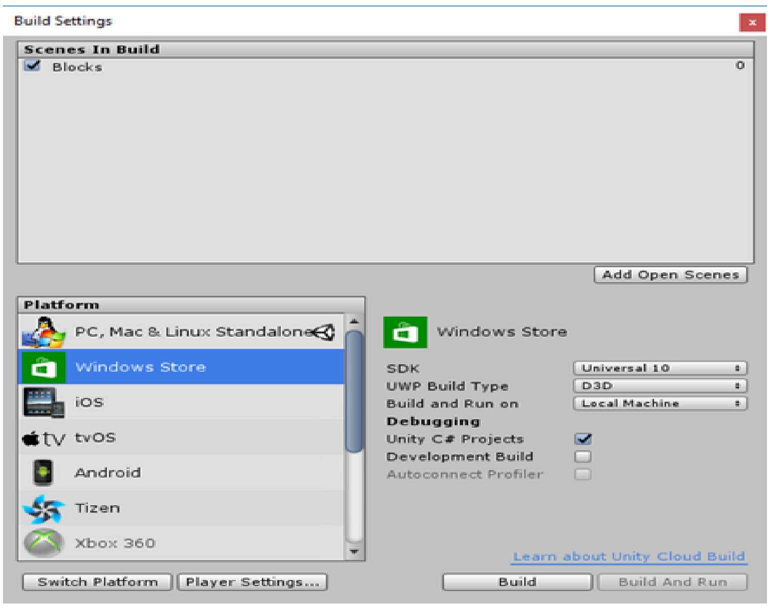

Fig 3.7 Build Settings in Unity

Fig 3.7 shows the build settings in Unity. The app has to be built each time any changes are made in Unity, so that a solution file will be developed, which when run using Visual Studio gives us the modified app in Hololens.

Once done, Click on **Build** button and select a folder (Create it for first time) where you want to keep your app code.

Once Selection done, Unity will start building the solution automatically and will create necessary player and ***Visual Studio Solution.***

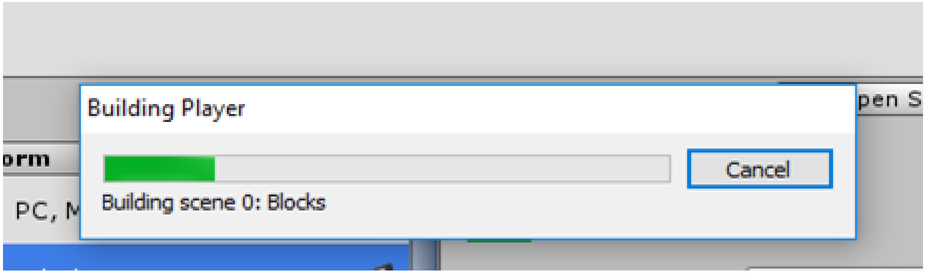

Fig 3.8 Building the Packages

Fig 3.8 shows the building of the packages. This is what occurs after Build settings are selected in Unity.

Once Unity done with his job, you will have the code folder open in front of you with all necessary files.

Later, a file with .sln extension is obtained. Open that file which opens the visual studio.

To run the application in the HoloLens Emulator, choose following option

- Mode – **Release**
- Platform – **x86**
- **HoloLens Emulator**

Fig 3.9 HoloLens Emulator

Fig 3.9 shows the Hololens emulator. This can be used when the Hololens is not present to test on. It is a software which emulates what the user would see on the Hololens. It allows the user to test holographic apps on your PC without a physical HoloLens.

3.2.3 Transfer to HoloLens

The entire application after viewing in HoloLens emulator can then be put into the HoloLens device by changing the run option from **HoloLens Emulator** to **Local Machine.** The device needs to be connected to the PC through USB cable and then the run option needs to be clicked. After clicking the run option, the app is installed in the device and it can be run as per the user. The user can then unplug the USB cable, and reopen the application from the start menu. By doing so, he transfers the updated application to the Hololens, and can use it untethered by a cable.

3.2.4 Execution of Application

The application is installed on the HoloLens Device. After the application is opened, different objects can be viewed. By walking around, the user can walk inside any object he wishes to examine. He can then use the AirTap gesture to select any part of a hologram. This gesture will make a menu appear considering of options to either adjust the hologram or remove it. It is shown in Fig 3.10.

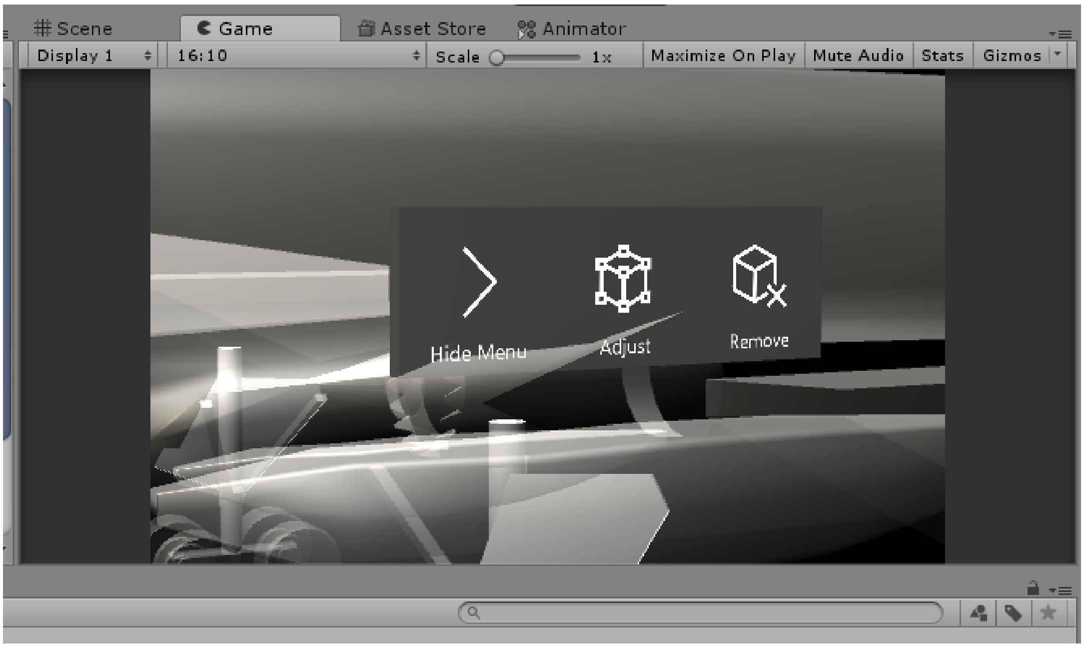

Fig 3.10 AppBar Menu

Fig 3.10 shows the AppBar menu which will be displayed when a hologram is selected by the user using an AirTap gesture. It shows the different things the user can do with the hologram.

If the user chooses the adjust option, a bounding box appears around the hologram. It is shown in Fig 3.11. By choosing any node, the hologram can be resized and rotated. If any part of the hologram other than the node is selected, it can be used to move the hologram around by a push action. Similarly, all the holograms in the given scene can be modified.

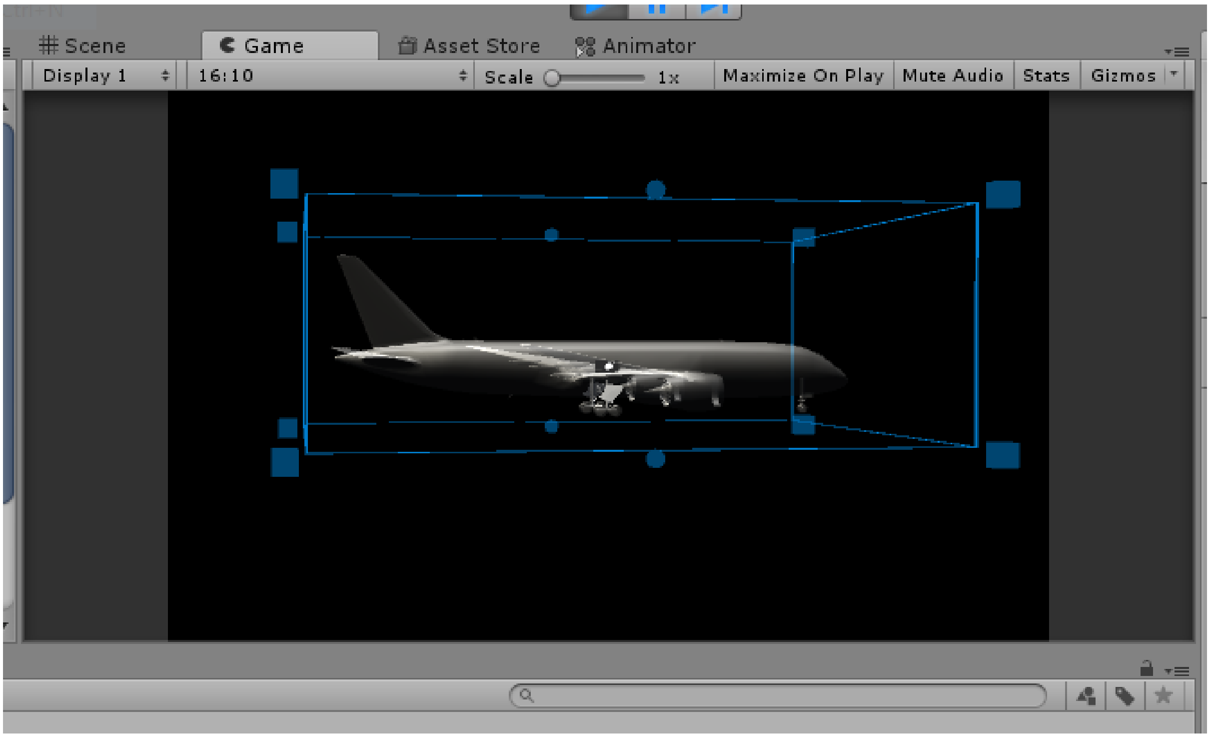

Fig 3.11 Bounding Box

Fig 3.11 shows the bounding box, which appears after the Adjust button is selected by the user from the AppBar menu. Using this bounding box, the user can scale, move or rotate the hologram as needed.

3.3 System Requirements

3.3.1 Hardware Requirements

1. Microsoft HoloLens

2. Minimum of 8GB RAM PC or Laptop

3. Intel Core i5/i7

3.3.2 Software Requirements

1. Operating System : Windows 10 Professional or Enterprise Edition

2. Unity 2017.1.0p

3. Microsoft Visual Studio 2015

4. Blender 3D Design Software

5. HoloLens Emulator (Hyper - V)

6. Coding Language: C#

4. Results and Discussions

Below are the screenshots of the output taken from the HoloLens device. To take the screenshots, the application was deployed in a room where it was free to move about the holograms.

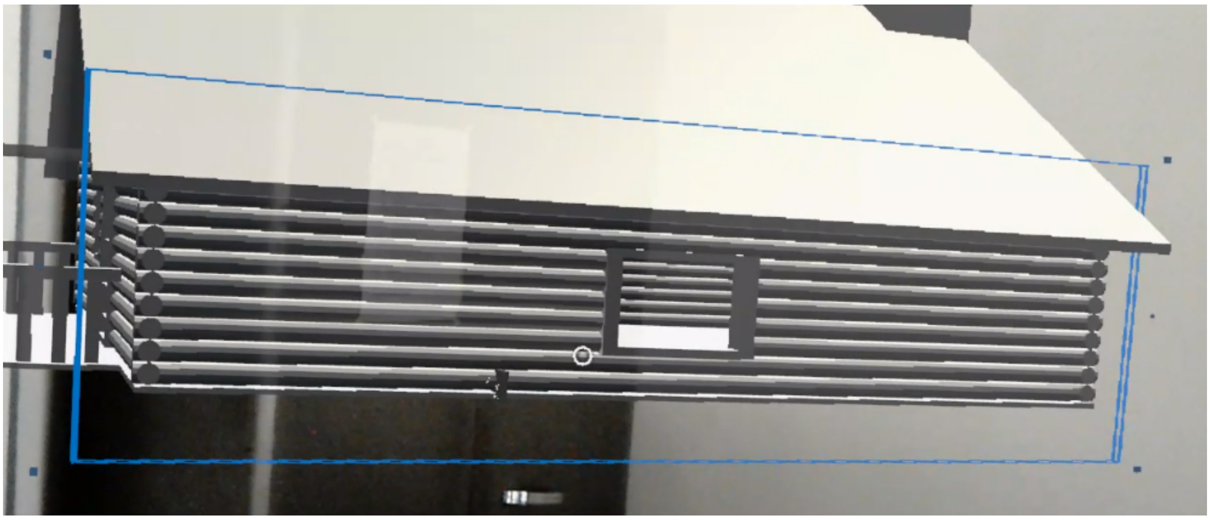

Fig 4.1 Hologram of a House

Fig 4.1 displays the hologram of a house deployed on the device. This is done by modelling the house model in Blender, after which it is exported into Unity. In unity, it is added as an asset and then as a GameObject into the Scene. The house is then given different features by adding Components to it. The placement of the hologram is also done in Unity.

The application is then built, and then run on the Hololens. It will display the hologram as above.

The next step is to select a part of the hologram which needs to be modified. When the hologram is selected using an AirTap gesture (similar to pinching the air with the hand out stretched). This will display a Menu, showing three buttons, namely- Hide Menu, Adjust and Remove.

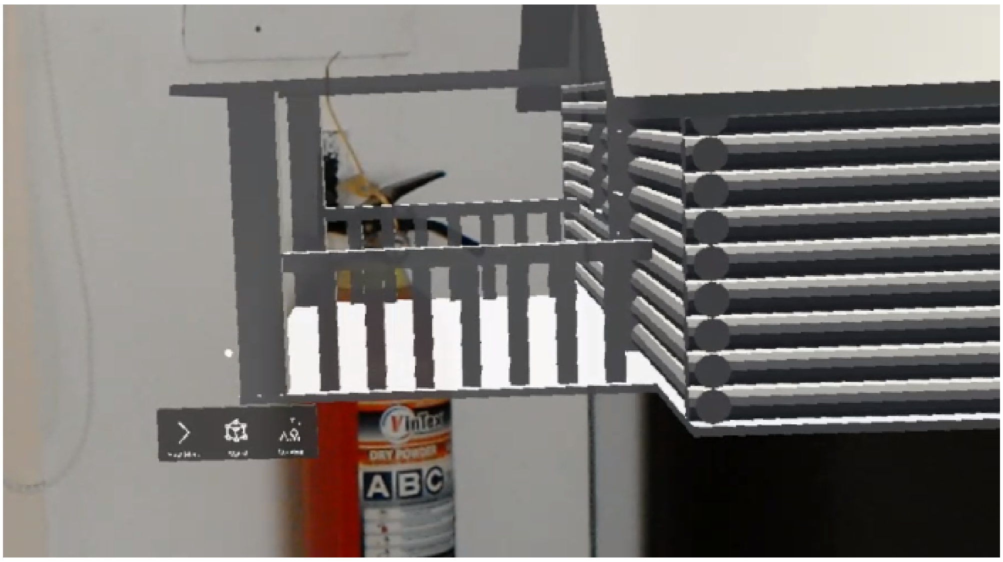

Fig 4.2 App Bar

Fig 4.2 shows selecting the porch railing which displays an app bar- showing a menu consisting of Adjust and Remove buttons. This enables the railing to be modified as needed by the user.

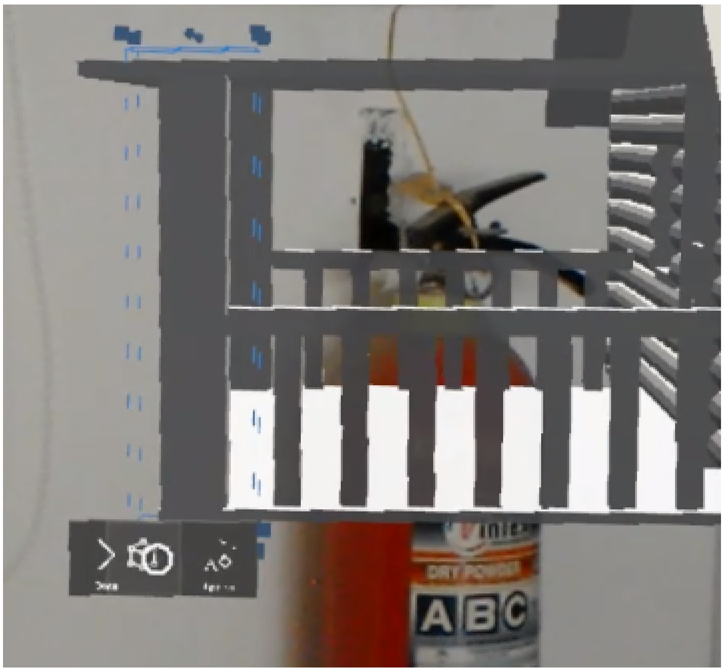

Fig 4.3 Bounding Box

Fig 4.3 shows what happens when Adjust button is selected. A bounding box appears around the porch railing, which essentially shows a boundary of the object selected. The edge nodes can be used to scale the railing by using Tap and Hold gesture and pulling the node where required. We can also translate the selected hologram by touching a part of the hologram and moving it around while still holding the hologram. Rotation can also be perfomed by touching an edge node and turning the hologram in the direction we need it to. Hence, bounding box can be used to make modifications to the selected hologram.

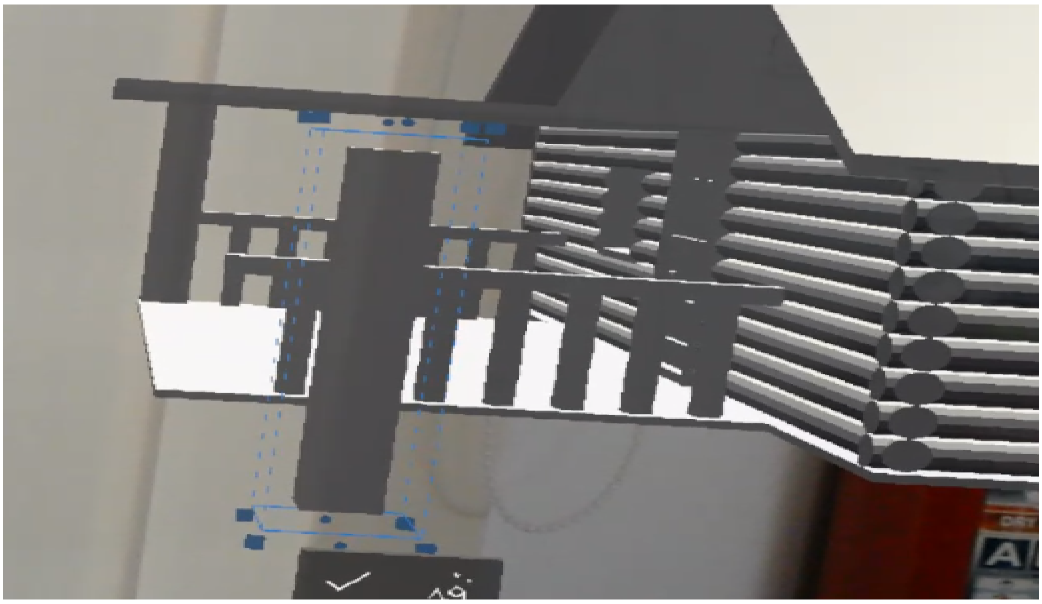

Fig 4.4 Move Operation on a Hologram

Fig 4.4 shows the translated porch railing. The move operation is performed by selecting Adjust button from the AppBar, which produces a bounding box. This bounding box is then used, by selecting any part of the hologram, and using it to drag the hololens around as needed. By using Tap-Hold-Drag gesture, move operation can be performed on the hologram.

This operation is useful because it helps the user get a feel of the surroundings as well as lets him understand if he would like to modify the object. He can do this by placing it in different positions, such as moving a wall farther away or closer to resize a room.

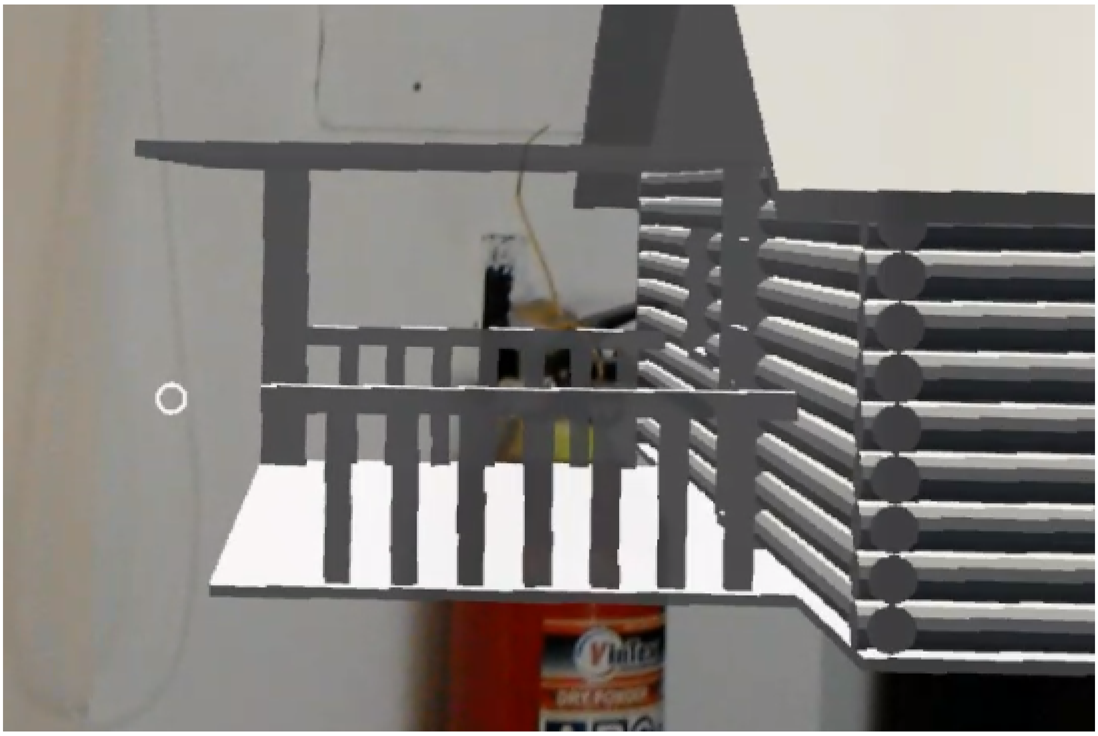

Fig 4.5 Remove Operation on a Hologram

Fig 4.5 shows the removal of the hologram which can be done if the remove button is selected from the AppBar. This can be used to delete a hologram if the user wants to see how the surroundings look without it.

An example of how this operation can be used is if the user wants to see how the room would look like if two adjacent rooms were merged. He can just delete the wall between them and see the outcome.

5. Conclusion and Future Work

It has been demonstrated effectively that an application can be built on the concepts of mixed reality, virtual interactions and reactions to various inputs. With respect to this application, the main motive of design is to understand and utilise various input mechanisms of the HoloLens and use them to manipulate various objects. Our approach leveraged the fundamental idea of interactions between the objects based on different gestures, and the manipulations that can be performed.

In the future, the application can be improved by giving features such as the ability to change colours of different objects. It can also have the ability to add or remove furniture by having menus to choose from. The application could be modified so as to include the option to take screenshots of what the user is viewing within the app. Object rendering can be made smoother by allowing complex models to be viewed properly.

References

[1]https://hololens.reality.news/how-to/hololens-dev-101-get-started-creating-for-windows-holographic-0174598/

[2]https://jeremylindsayni.wordpress.com/2016/08/28/coding-for-the-hololens-with-unity-5-part-7-creating-a-basic-hud/#post-8886

[3]https://docs.microsoft.com/en-us/windows/mixed-reality/academy

[4]https://docs.microsoft.com/en-us/windows/mixed-reality/holograms-210

[5] https://docs.microsoft.com/en-us/windows/mixed-reality/holograms-211

[6] J. C. Lee, S. E. Hudson, J. W. Summet, P. H. Dietz, "Moveable interactive projected displays using projector based tracking", 18th annual ACM symposium on User interface software and technology, pp. 63-72, 2005.

[7] S. Pessoa, G. Moura, J. Lima, V. Teichrieb, J. Kelner, "Photorealistic rendering for augmented reality: a global illumination and BRDF solution", IEEE-VR2010 IEEE, pp. 3-10, 2010.

[8] J. M. Teixeira et al., "miva: Constructing a wearable platform prototype", IX Symposium on Virtual and Augmented Reality Petrópolis, pp. 68-76, 2007.

[9] J.P.S.M Lima et al., "Applications in engineering using augmented reality technology", XXIX Iberian Latin American Congress on Computational Methods in Engineering Maceió, 2008.

[10] T. Hosokawa, S. Jarusirisawad, H. Saito, "online video synthesis for removing occluding objects using multiple uncalibrated cameras via plane sweep algorithm", Third ACM/IEEE International Conference on Distributed Smart Cameras 2009., pp. 1-8, 2009.

[11] Braunstein M. And Andersen, G. And Riefer D. 1982. The use of occlusion to resolve ambiguity in parallel projections. In Perception and Psychophysics, 31(3), 261-267

[12] Codella C. (1992) Interactive Simulation in a Multi-Person Virtual World. In Proceedings of CHI ’92. ACM Press, New York.

[13] Cutting, J. 1997. How the eye measures reality and virtual reality? In Behaviour Research Methods, Instruments & Computers, 29(1), 27-36

Appendix

A.1 AR, VR and MR

Augmented reality (AR) is a live, direct or indirect view of a physical, real-world environment whose elements are augmented by computer-generated sensory input such as sound, video, graphics or GPS data.

Augmented reality is an overlay of content on the real world, but that content is not anchored to or part of it. The real-world content and the CG content are not able to respond to each other.

Fig A.1.1 Augmented Reality - Pokemon Go

Fig A.1.1 shows Augmented Reality being used in the game Pokemon Go, where Pokemon (computer generated graphics) are augmented into our surroundings (real world).

Mixed reality (MR)—sometimes referred to as hybrid reality—is the merging of real and virtual worlds to produce new environments and visualisations where physical and digital objects co-exist and interact in real time.

Mixed reality is an overlay of synthetic content on the real world that is anchored to and interacts with the real world—picture surgeons overlaying virtual ultrasound images on their patient while performing an operation, for example. The key characteristic of MR is that the synthetic content and the real-world content are able to react to each other in real time.

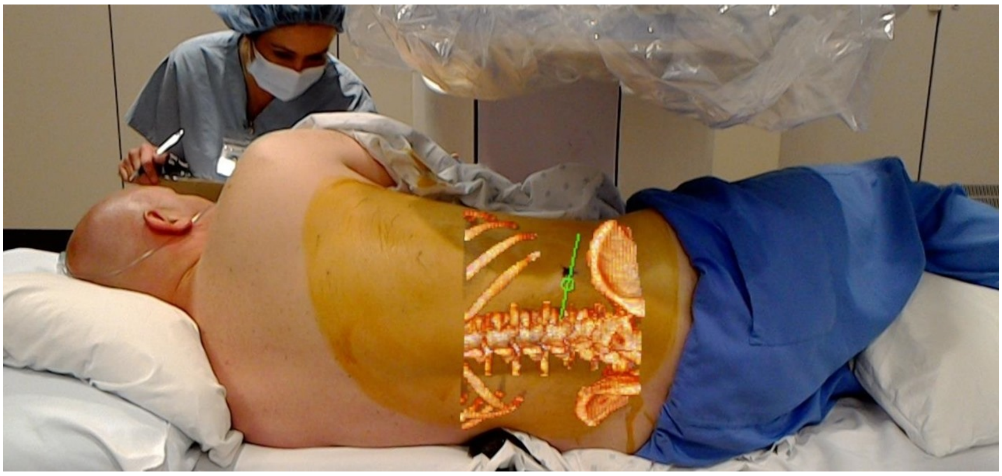

Fig A.1.2 Mixed Reality - Surgery

Fig A.1.2 shows Mixed Reality(in the form of Hololens) being used by surgeons while performing open spine surgery.

Virtual reality (VR), which can be referred to as immersive multimedia or computer-simulated reality, replicates an environment that simulates a physical presence in places in the real world or an imagined world, allowing the user to interact in that world.

Fig A.1.3 Immersive Virtual Reality

Fig A.1.3 shows the usage of Virtual Reality in which the user is in an entirely imagined world. VR can be used for immersive experiences.

A.2 Gestures

- AirTap gesture **-** Air tap is a tapping gesture with the hand held upright, similar to a mouse click or select. This is used in most HoloLens experiences for the equivalent of a "click" on a UI element after targeting it with Gaze.

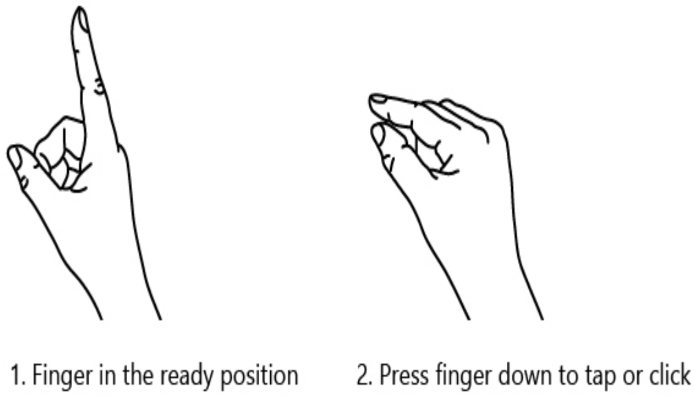

Fig A.2.1 AirTap gesture

Fig A.2.1 shows the AirTap gesture in which finger is initially in ready position and then a tap or click motion is performed to select.

- Bloom gesture - Bloom is the "home" gesture and is reserved for that alone. It is a special system action that is used to go back to the Start Menu. It is equivalent to pressing the Windows key on a keyboard or the Xbox button on an Xbox controller. The user can use either hand.

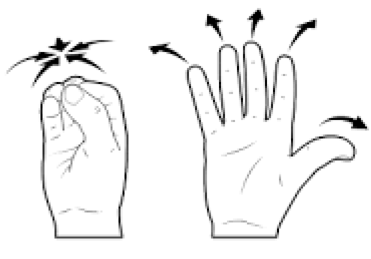

Fig A.2.2 Bloom gesture

Fig A.2.2 shows the Bloom gesture in which the user has to hold out his hand, palm up, with his fingertips together and then open his hand.

A.3 Important Terms

- CAD files - CAD (computer-aided design) software is used by architects, engineers, drafters, artists, and others to create precision drawings or technical illustrations. CAD software can be used to create two-dimensional (2-D) drawings or three-dimensional (3-D) models.

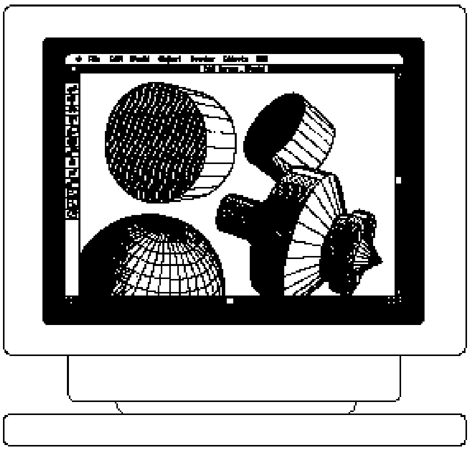

Fig A.3.1 3-D CAD file

Fig A.3.1 shows a three dimensional CAD file. In the CAD Explorer application, such files can be viewed through Hololens.

- Blender - Blender is a professional, free and open-source 3D computer graphics software toolset used for creating animated films, visual effects, art, 3D printed models, interactive 3D applications and video games.

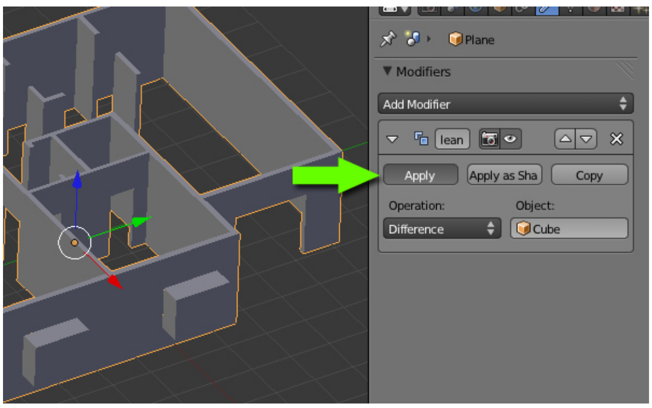

Fig A.3.2 Blender object creation

Fig A.3.2 shows the Blender software being used to create 3D model of a house, similar to what is done in the project.

▪ Unity - Unity is a cross-platform game engine developed by Unity Technologies. and used to develop video games for PC, consoles, mobile devices and websites. It supports three languages: C#, UnityScript (also known as JavaScript for Unity) and Boo. In this project, we made use of C# scripts to write the application.

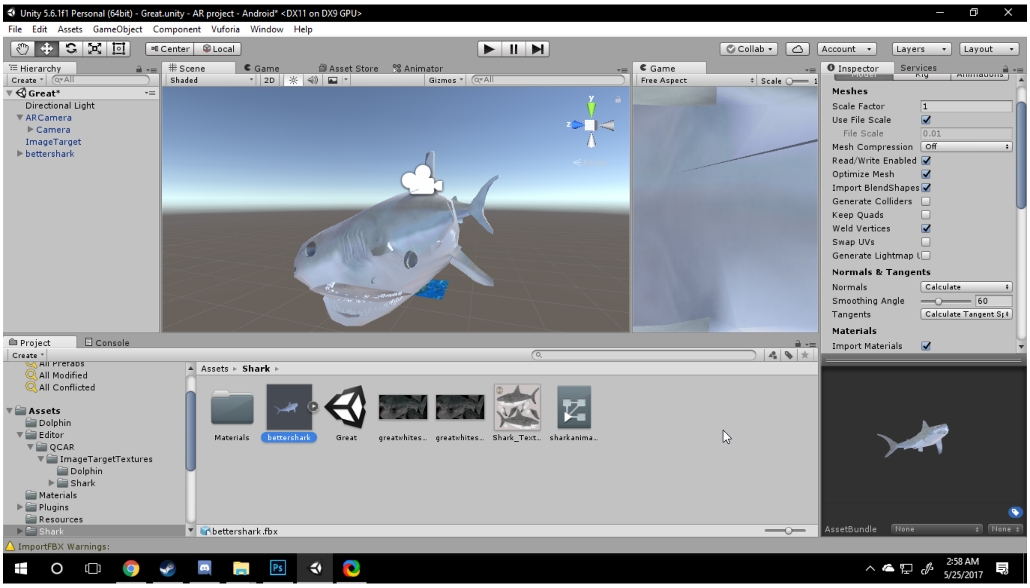

Fig A.3.3 Unity

Fig A.3.3 shows using of the Unity software, in order to add different specifications that need to be shown in the application.