Design and Development of an Image Analysis Based Interaction Controller for Mobile VR Edutainment Application

Safiul Kabir

Aniruddha Prithul

Computer Science and Engineering Discipline,

Khulna University

Khulna-9208, Bangladesh

February, 2017.

Design and Development of an Image Analysis Based Interaction Controller for Mobile VR Edutainment Application

Safiul Kabir

Student ID: 120211

Aniruddha Prithul

Student ID: 120224

Computer Science and Engineering Discipline,

Khulna University

Khulna-9208, Bangladesh

February, 2017.

KHULNA UNIVERSITY

COMPUTER SCIENCE AND ENGINEERING DISCIPLINE

The undersigned hereby certify that they have read and recommend to the computer Science and Engineering Discipline for acceptance a thesis entitled "**Design and Development of an Image Analysis Based Interaction Controller for Mobile VR Edutainment Application**" by **Safiul Kabir (ID: 120211)** and **Aniruddha Prithul (ID: 120224)** in partial fulfillment of the requirements for the degree of **Bachelor of Science in Computer Science and Engineering (CSE)**

Dated: 7th February, 2017

Kazi Masudul Alam Supervisor
Assistant Professor
Computer Science and Engineering Discipline
Khulna University

Dr. Md. Rafiqul Islam Second Examiner
Professor
Computer Science and Engineering Discipline
Khulna University

Dr. Kamrul Hasan Talukder Head of the Discipline
Professor
Computer Science and Engineering Discipline
Khulna University

Acknowledgement

First of all, we thank the Almighty Allah, the most gracious, the most merciful, for giving us the intelligence and ability to complete our thesis.

We like to express our deepest gratitude to our supervisor Kazi Masudul Alam for encouraging us to think and do something out of the box. Without his continuous cooperation, suggestion, observation and inspiration, we couldn't reach our goal. He spent a lot of time with us to improve this thesis further. He always encouraged us to make the smallest of adjustments that would make our work more efficient.

Our second examiner, Dr. Md. Rafiqul Islam, has been always there to listen to us and give advices. We are deeply grateful to him for carefully reading and commenting on multiple revisions of this manuscript. Also, long discussion with him allowed us to further improve existing features and add more dimensions to our work.

Moreover, we would like to express our gratitude to all our respected teachers of CSE Discipline for their goodwill support and advice. We also thank our friends and seniors for their heartiest cooperation and encouragement that were very much helpful to us.

Abstract

Virtual Reality (VR) is the idea of placing its user in a virtual world. Our experience of reality is a combination of sensory information. Virtual Reality devices provide us with this information and allow us to be immersed in a virtual world. Edutainment is a technique of entertaining people while educating. This technique can be used in VR also. In recent times, Virtual Reality devices have been improved to a great extent. Some of these devices consist of Head-Mounted Displays (HMDs) along with controllers with positional tracking capabilities to allow the user interact with the virtual world. Using these controllers, the user can manipulate virtual objects in the virtual world by moving his/her hands in the real world. Some of the problems associated with these devices are: they work with specific VR systems, they require high-end machines to do processing and total cost of a complete set is still high. All these factors reduce the accessibility of VR. In this work, we propose an interaction controller that works readily with mobile VR systems like Google cardboard, Gear VR etc. and increases the effectiveness of virtual reality. It is simple in design. We've also developed a smartphone application to demonstrate the use of the controller.

Table of Contents

List of Figures

List of Tables

CHAPTER 1

Background

1.1 Introduction

The meaning of 'virtual' is near and reality means our experience as a human being. So 'Virtual Reality' means 'near-reality'. The ideas behind virtual reality are based on theories about human desire to cross the boundaries of the 'real world' by creating a virtual world. In that world, we can interact with a virtual environment which will enable new classes of human-machine interaction (HMI).

The common forms of HMI such as a keyboard, mouse, and monitors etc. force people to adapt to the demands of the technology. But the other way around is desirable. This alternative approach that allows someone to immerse themselves in a virtual world by means of their senses is known as virtual reality. The fabricated environment that gives the feelings of the real environment is known as a virtual environment. This new form of interaction often results in more effective and dynamic forms of communication and understanding.

The method by which virtual reality headsets operate is not much complex. The main principle is – it captures the user's physical movements in the real world, then a rendering computer redraws the virtual world to reflect these movements. The updated representation of the virtual world is presented to the user in the real world via a head mounted display. In this process, the user feels as if he is in the virtual world himself as all he sees is his rendered movements in the virtual world.

Now, the movement tracking process can be accomplished in many ways. One of those ways is to give the user a controller in his hand and track that controller.

1.2 Motivation

A variety of devices has been used to replicate the human movements of the real world as inputs or control signals in the virtual world. These devices provide some advantages as well as limitations. Almost all of these have some powerful sensors as well as some connection mechanism with the VR device itself. They send various sensor data to the VR device and the

VR device calculates them in order to replicate the movement information to the virtual world. This technique has some overhead and requires a significantly high capacity power source because of the involvement of the sensors and connection module i.e. Bluetooth, IR etc. This is why we were encouraged to do this research to design and develop an alternative approach that requires relatively low power and cost, still providing similar if not better performance.

1.3 Objective

Most of the current virtual reality gears require high budget and high power machine. Our objective is to develop a solution that enables the user to get immersed in the world of VR by using a mobile VR headset, a vision based controller, a processing unit with a camera and a smartphone that runs the VR application. The controller itself uses no sensor and there is no wired or wireless connection between the controller and the VR device.

1.4 Research Contribution

- ✓ A vision based HMI controller that can be used with mobile VR systems. The controller is similar to a torch light. The movement of the hand holding the controller is captured by a camera and is replicated in the VR world inside the application.
- ✓ An edutainment VR application. The application has three different mode of play. In first and second mode, the user has to build structure using square blocks. In the third mode, the user can paint in a VR environment with the controller acting as a paint brush.

1.5 Thesis Organization

In the next chapter, chapter 2, we have shown a summary of the history and evolution of edutainment, virtual reality, and VR devices. Also, the current systems, which are available for experiencing virtual reality and edutainment, are discussed. In chapter 3, we describe our edutainment system architecture explaining different modules of the system. Chapter 4 describes tools and technologies, edutainment application features and interaction, and limitations of our system. And the last chapter, chapter 5, contains the conclusion with an overview of our developed system with possible future work ideas.

CHAPTER 2

Literature Review

2.1 Background

Virtual reality is a three-dimensional, computer generated environment where a person can explore and interact with. That person becomes part of this virtual world or is immersed within this environment and whilst there, is able to manipulate virtual objects or perform a series of actions. [1]

Edutainment is the mix of education and entertainment. It can be seen as both educational content with entertainment value and entertaining content with educational value. Education is a must for people in a society. At the same time, education alone is not very interesting. So, unless education has some fun in it, it is very easy to get carried away. Here comes the idea of edutainment that helps education being entertaining so that there be no lapse of concentration. [2]

We can see edutainment as a process of subconscious learning. And VR enables an immersive way of learning for the user. So, mixing these two concepts, we can expect a more effective result.

2.1.1. History of Virtual Reality

Virtual Reality is a relatively new concept in the field of computer science. The concept was first expressed in 1930, in a story by science fiction writer Stanley G. Weinbaum titled "Pygmalion's Spectacles". In the story, the writer described the idea of wearing goggles that is very much similar to modern VR headsets. This makes him a true visionary of the field.

Figure 1: Advertisement for Heilig's Sensorama

Sensorama (patented 1962) was an arcade-style theater cabinet that was developed by cinematographer Morton Heilig in the mid-1950s. This would stimulate all the senses. It featured stereo speaker, a 3D display, smell generators, fans and a vibrating chair. The machine was intended to immerse the user fully in the film. He also created six short films for the machine. [3]

Morton's next invention was the Telesphere Mask (patented 1960) and was the first example of a head-mounted display (HMD), though the non-interactive film medium lacked any motion tracking. The headset provided stereoscopic 3D and wide vision with stereo sound. [3]

Next came the idea of "Ultimate Display". It was described by Ivan Sutherland. The idea was to simulate reality to the point where the actual reality couldn't be differentiated. This concept included viewing a virtual world through a head-mounted display with augmented 3D sound and tactile feedback.

"*The ultimate display would, of course, be a room within which the computer can control the existence of matter. A chair displayed in such a room would be good enough to sit in. Handcuffs displayed in such a room would be confining, and a bullet displayed in such a room would be fatal. With appropriate programming, such a display could literally be the Wonderland into which Alice walked*." – **Ivan Sutherland** [4]

In 1993, Sega announced the Sega VR headset [5]. They showed a prototype that had head tracking, stereo sound, and LCD screens. For technical difficulties, this prototype never came out as a commercial device.

In 1999, a movie titled "The Matrix" by the Wachowski siblings' was released. In the movie, many characters were living in a simulated world without even knowing that it was not a real world. This movie brought the idea of simulated reality into the mainstream.

Since the beginning of the 21st century, there has been major and rapid development of virtual reality. High-end smartphones with high-density screens have enabled the mass people to get the taste of virtual reality. This made the companies very much interested in this field. This is why we're seeing amazing technologies using the concept of simulated reality in our everyday life.

2.1.2. Edutainment

Edutainment is a process that makes the learning process entertaining. It is, however, not a recent discovery. The technique is being used for a long time. Modern edutainment tools and mediums use a mixture of audio and video. This includes films, TV productions, video games, museum exhibits, radio, toys and more. Since the 1970s, various groups in the western countries have used edutainment to obtain more effective result in various sectors. [2]

Figure 2: The Oregon Trail Screenshot from the Apple II Version [6]

"**The Oregon Trail**" is one of the earliest computer games that was originally designed to teach school children about the realities of 19th-century pioneer life on the Oregon trail. The role of the player is of a wagon leader who guides his party of settlers from Independence, Missouri, to Oregon's Willamette Valley on the Oregon Trail via a covered wagon in 1848. This game became so successful that it was inducted into the **World Video Game Hall of Fame.** [6]

"**Odell Lake**" is another early educational life simulation game. It was produced by MECC. In this game, the character is a fish living in the Odell Lake which is a real-world lake in Oregon. There are two modes of play in the game: "go exploring" and "play for points". The user has to decide which fish to eat to survive and avoid other enemies. [7]

Figure 3: Odell Lake gameplay (Apple II)

2.2 Related Works

Virtual Reality (VR) hardware and software development is on the move at present. There are many devices which are on the prototypical state and not yet commercially available.

However, to categorize the VR hardware, we can divide these into two categories: input devices and output devices. In most cases, VR uses a Head-Mounted Display (HMD) that provides output in its screen and also provides input data.

We can categorize the HMDs into two categories: *wired* and *mobile*. The mobile HMDs are wireless and doesn't require an additional PC. In most cases, interaction in the virtual world using a mobile HMD is very much limited to displaying 360^{o} videos or panoramas.

The feature list of wired HMDs is a large one. They provide more quality than mobile HMDs in terms of resolution, field of view and weight. They are more complex in design. Some of wired HMDs are equipped with cameras, some provide 6 degree of freedom (DOF) tracking system. The drawback is that most of them has to be connected to a powerful PC. Also the VR systems themselves are expensive [8].

In the next sub-section, we've shown some of wired HMDs, mobile HMDs, VR input devices and VR controllers.

2.2.1 Wired HMDs

One of the most common images that comes to our mind when thinking about VR is of someone wearing goggles like devices on the head covering the eyes. The Head-Mounted Display (HMD) is a hybrid device that provides both input and output for virtual environment. There are many HMDs currently available in the market. Most of them have stereoscopic displays with head tracking systems. This enables the user to view 3D images with a big field of vision and have the point of view move according to the user's head position and orientation. There is one display for each eye and the stereoscopic images are created simply by projecting two different images of the same subject taken from two viewpoints slightly apart. Commonly, gyroscope and accelerometers are used to track the head orientation of the user.

Oculus Rift

Figure 4: Oculus Rift Development Kit 2

The Oculus Rift is a headset built with the main focus on gaming. It provides stereoscopic vision and fast head tracking. It accomplishes this by processing data that comes through a 3-axis gyroscope, an accelerometer and a magnetometer, giving the user a fast image update, meaning no noticeable delay. In Oculus Rift Development Kit 2, the 1920 x 1080 HD resolution delivered a 960 x 1080 display to each eye; its refresh rate was 60Hz, and horizontal field of view was 100-degree [9].

HTC Vive

Figure 5: HTC Vive

HTC vive is a VR headset developed by *HTC* and *Valve Corporation*. It needs a powerful PC to run with. It is designed for room scale use. It uses laser tracking that allows the user to walk around a 15x15 foot room [10].

The refresh rate of the vive is 90 Hz. It uses two screens, each having a resolution of 1080x1200. The device uses more than 70 sensors. By March 2016, more than 100 games were known to be coming to virtual reality format for HTC vive [11].

PlayStation VR Headset

Figure 6: PlayStation VR Headset

PlayStation VR headset is a HMD developed by *Sony Interactive Entertainment*. It was designed to work with PlayStation 4 video game console. The PlayStation VR system can output a picture to both PlayStation VR headset and a television simultaneously. The headset has a 5.7-inch OLED panel, with a display resolution of 1080p. It also has a processor box, a 3.5mm audio jack and nine positional LEDs on its surface to track 360-

degree head movement using the PlayStation camera. It is capable of displaying content at 120fps with a field of view of 100-degree [12].

2.2.2 Mobile HMDs

Mobile HMDs carry a smartphone as a whole for display and processing of data. Google developed the first device of this kind. It is named *Google Cardboard* [8]. Mobile HMDs doesn't require an additional PC.

Google Cardboard

Google Cardboard is a virtual reality platform developed by Google. It is named for its fold-out cardboard viewer. Users can build their own viewer or purchase a pre-manufactured one. To use this, the user needs to place the phone into the back of the viewer and view content through the lenses. The smartphone needs to have gyroscope sensor to use cardboard [13].

Figure 7: Google Cardboard

Gear VR

Gear VR is a mobile virtual reality headset developed by *Samsung Electronics*. It has to be used with a compatible Samsung galaxy device. The device acts as the display and processor of the device. Field of view of the latest version of Gear VR headset is 101-degree [14].

Figure 8: Samsung Gear VR

2.2.3 VR Controllers

The high-end VR systems have controllers having 3D positional tracking capability. The mobile VR systems lacks this feature but have some controllers with some other features. In figure 9, we’ve shown controllers for (a) PlayStation VR, (b) HTC Vive, (c) Oculus Rift. These are some advanced controllers with multiple sensors and a large list of features. Each of them works only with its dedicated VR system. [10]

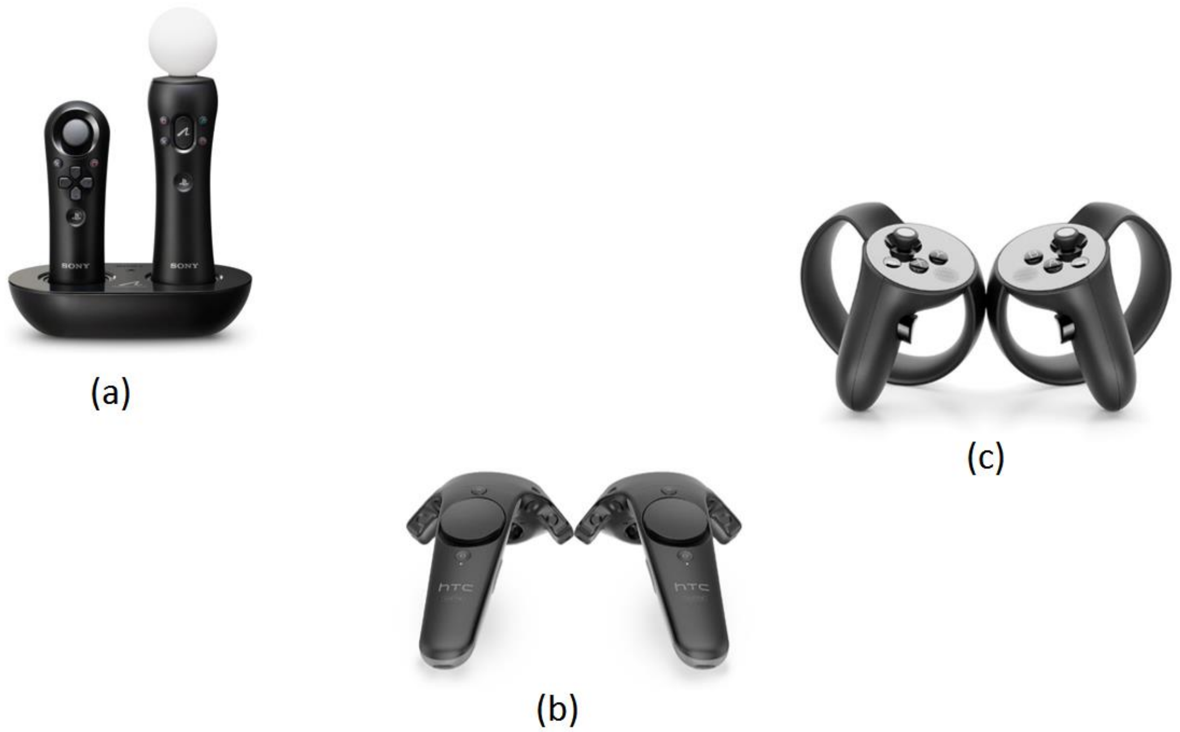

Figure 9: Controllers for wired HMDs, (a) PS move and navigation controller [28], (b) HTC Vive controller [11], (c) Oculus Rift controller [9].

2.2.4 Cave Automatic Virtual Environment

Cave Automatic Virtual Environment (CAVE) is a virtual reality room. Projectors cover the walls of the room with stereoscopic images and the user needs to use glasses which are synchronized with the alternating images the projectors and speakers are placed around the room to surround it with sounds. [15]

Figure 10: Visualization of a 3D model of a ship in CAVE

2.2.5 Input Devices

The usual approach to data input in VR is movement recognition. Having a device or a set of devices that reads and processes natural movement would change the interaction between the users with the virtual world to a more intuitive way. Here are the most common types of data input to VR.

2.2.3.1 Wired Gloves

Wired gloves can measure joint angles, pressure, tracking and haptic feedback. There are three main technologies used in wired gloves: using light through fiber optics, using conductive ink to measure electrical resistance, and using mechanical sensors. [16]

Figure 11: Wired Gloves

Gloves that use fiber optics measure finger flexion with the use of light and photo sensors. The angle of bending of the joints is determined by the amount of light received by each photocell, the more the user bends his fingers, the less light is captured by the sensors, and vice-versa.

Other gloves use an electric conductive ink and as the user makes movements, the electrical resistance changes and makes possible for a processor to identify movements. This technology does not have a performance as good as the fiber-optics gloves but is cheaper. This led a lot of VR enthusiasts to get a Power Glove, which uses this technology, to make their own software.

Another approach is the use of a Dexterous Hand Machine, which is more similar to an exoskeleton than to a glove, as mechanical sensors need to be fixated at the hand's joints. These are more precise than the other two technologies, at the price of being rather bulky to wear.

2.2.3.2 Wands

The Nintendo Wii was the first console that brought motion-sensor controllers in popularity. The Wii remote controller has an infrared sensor to identify the IR light emitted by a sensor bar placed at the top or bottom of the television, provided with five IR emitters in each side. The console calculates the position of the controller based on the distance of a fixed point and the point the Wii controller read the infrared signal. [17]

Figure 12: Nintendo Wii Controller

The controller is also equipped with an accelerometer which detects three axis movements and is sent to the console through Bluetooth. A gyroscope determines the inclination. It also has speakers and a simple rumble pack to give the user feedback. Sony's response to the Wii was the combination of the PlayStation Eye, which is a usual digital camera, and the PlayStation Move, the proper motion controller. The first thing one see the PS Move is the ball on the top of it. It is actually important for motion detection, as the PS Eye identify the ball to recognize movements. It possesses three LEDs (Red, blue and green), which make the orb change to a different color, thus easing the detection made by the PS Eye. Similar to the Wii controller, PS move is equipped with a three-axis accelerometer and a gyroscope to determine its orientation. In addition, a magnetometer, which measures the Earth's magnetic field and is used to calibrate inertial sensors on the controller.

2.2.3.3 Computer Vision

Using cameras to recognize models and identify motion, we can enable movement interaction. The most famous device using such technology is the Microsoft Kinect. It has been greatly explored by the industry, universities, and hobbyists since its launch through reverse engineering or later available SDK. Its main capabilities are using an RGB camera that allows

Figure 13: Microsoft Kinect 2

face recognition; a depth sensor that enables it to scan the surroundings; a built-in microphone to capture sound; and a microchip to track and recognize movements. [16]

2.2.6 VR Edutainment Applications

A popular VR edutainment application is "**Google Arts & Culture VR**" developed by Google. In this application, the user can view arts from museums around the world by stepping inside a virtual gallery and see works by artists like Van Gogh or Rembrandt. The user can zoom into every brushstroke. He/she also can hear audio guides from expert museum curators. [18]

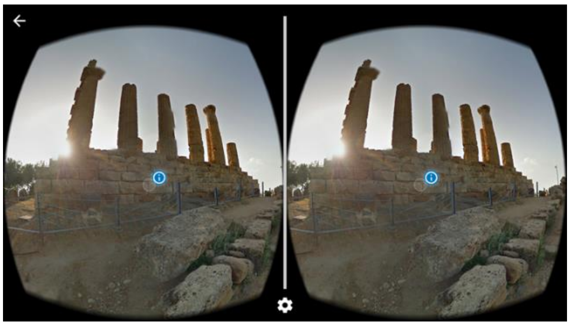

Figure 14: Screenshot from Google Arts & Culture VR

Another great VR edutainment application is "**Expeditions**" that is also developed by Google. This is a VR teaching tool that lets the user lead or join virtual trips all over the world or even outer space. For classroom use, this application allows a teacher to be a "guide" leading a group of "explorers" through collections of 3D and 360º images. [19]

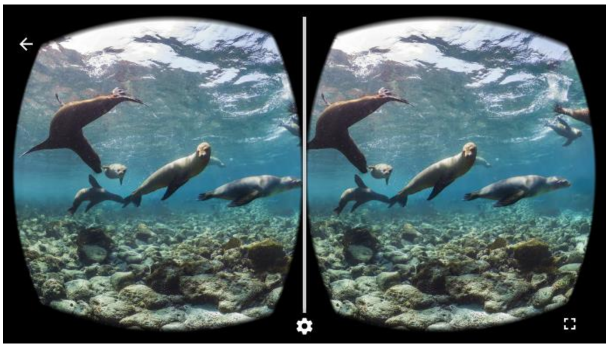

Figure 15: Screenshot from Expeditions

2.3 Discussion

In this chapter, we've given a brief summary of the current systems that allow users to get the experience of virtual reality. Most systems use Head-Mounted Displays that covers the eyes of the user with a display. Another good example that allows complete immersion is the Cave automatic virtual environment. The drawback of this is the need of a whole room. There are some input devices like wired gloves, wands and computer vision which are also discussed. The VR edutainment applications provide a very interesting way of learning.

Recently the companies are exploring this area very seriously. The table below shows the comparison between the current VR systems:

	Google Cardboard	**Gear VR**	**HTC Vive**	**Oculus rift**	**PlayStation VR**	**Proposed System**
Type	Mobile	Mobile	Tethered	Tethered	Tethered	Mobile
Platforms	Most Smartphones	Samsung Flagship Phones	PC	PC	PlayStation 4	Most Smartphones
Controller	Single Button	Trackpad	Primarily motion controllers	Xbox One gamepad	DualShock 4 and PS Move	Connectionless Interaction Controller
Positional Tracking	None	None	Laser Towers	Camera	Camera	Camera

Table 1: Comparison between proposed system and currently available commercial VR systems [10]

CHAPTER 3

Edutainment System Architecture

3.1 System Architecture

The edutainment system that we have developed is a system to let the users dive into a virtual world where they can make any structure using blocks. It also incorporates a game mode that exercises the user's memory by introducing a memory game. To do this, we've used a controller to detect the motion of the hand of the user. The motion is captured in real-time using a webcam. The webcam data is fed to a processing unit where the motion of the ball is detected by analyzing the received data. This motion data is then wirelessly sent to the smartphone where the application runs. Finally, the application replicates the motion made by the controller using the data received from the processing unit. The system architecture is shown in figure 11.

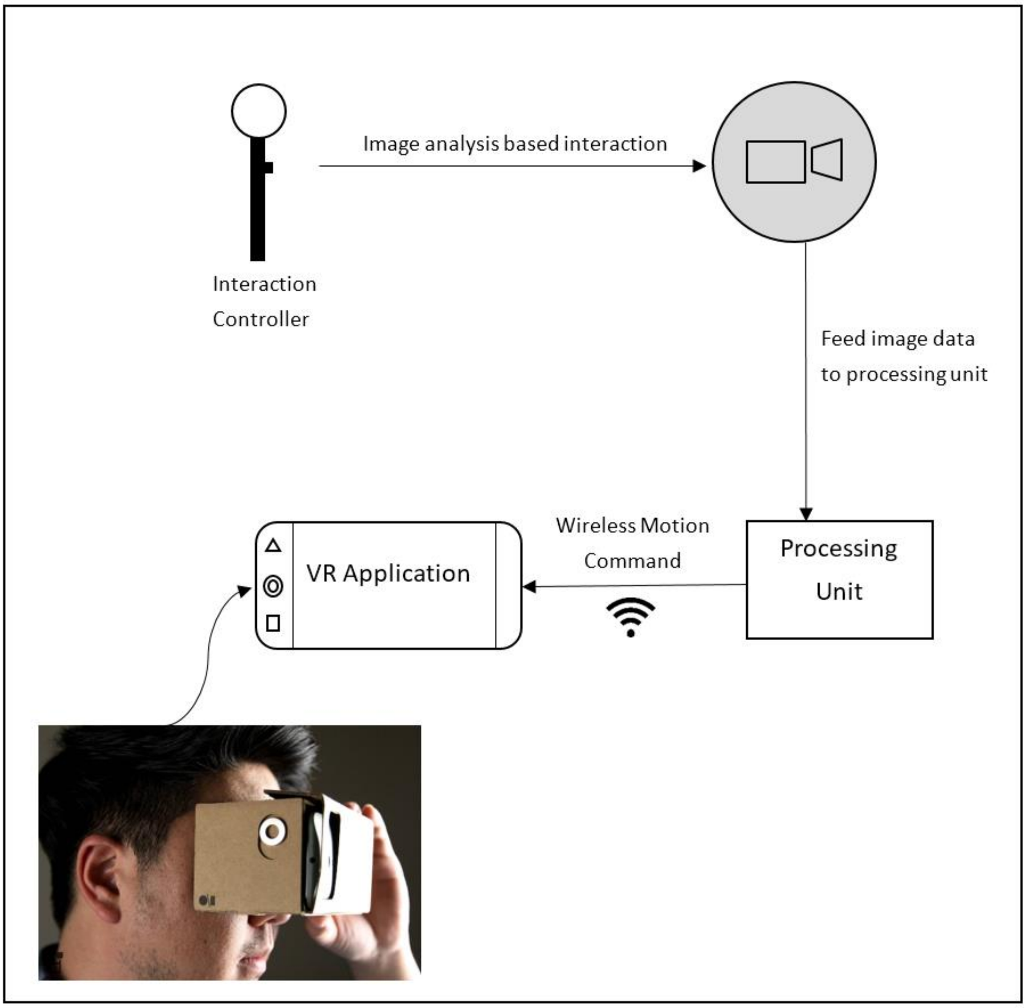

Figure 16: System architecture of our VR edutainment system

3.2 The VR Interaction Controller

The controller is very simple in design. The components of the controller are:

- A ping-pong ball
- Single RGB LED Bulb
- A battery (3 V)
- A resistor (150 Ω)
- A switch
- A casing to hold the circuitry

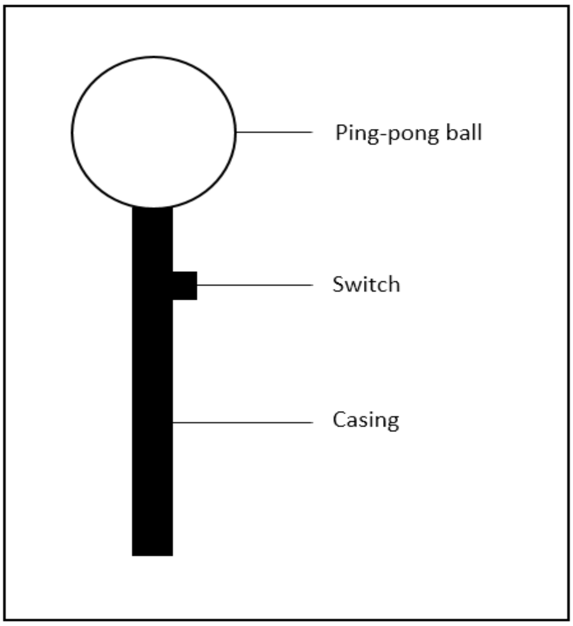

Figure 17: Diagram of the controller as envisioned

Figure 17 shows the controller as envisioned. It has an RGB LED enclosed inside a ping-pong ball. We also have the switch that makes the LED blink on soft press and turns on or off on hard press.

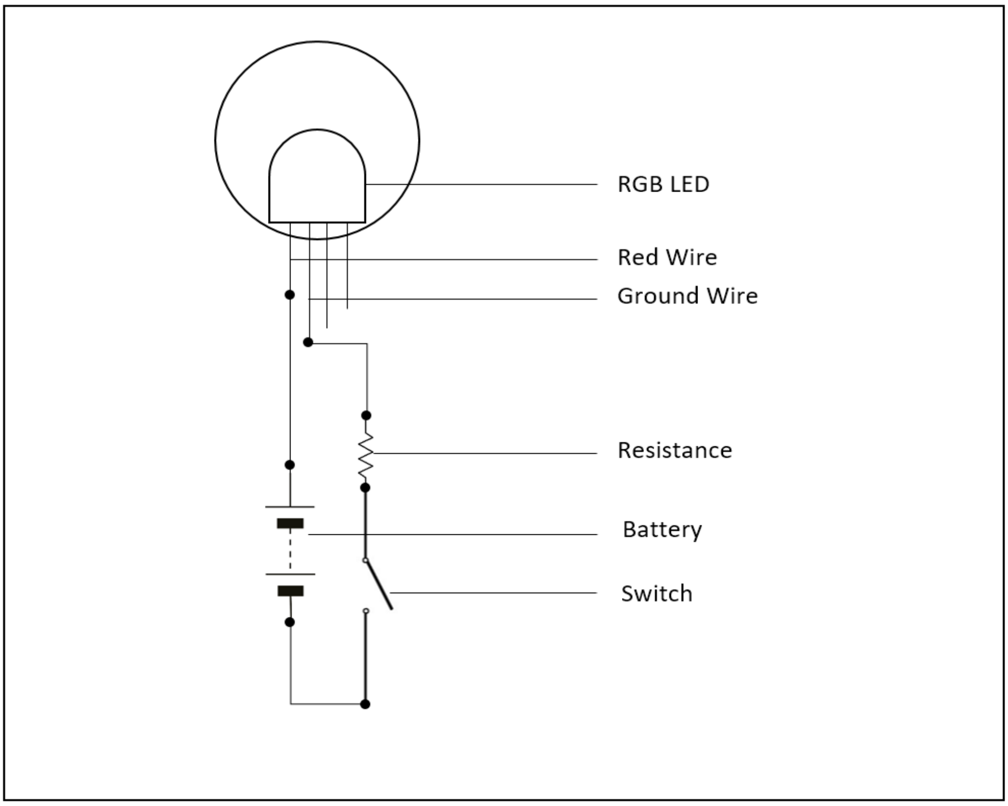

Figure 18: Circuit Diagram of the interaction controller

In figure 18, we have shown the circuit diagram of the interaction controller. We also have shown a photograph of the controller in figure 19.

Figure 19: Photograph of the interaction controller

3.3 Command Detection Process

The command detection process was carried out in two parts. In the first part, we determined the 2D position and radius of the orb from the image. In the second part, we've used this information to find the depth of the orb from the camera.

3.3.1 Determining Color Range

To detect the controller, we had to determine the range of the orb in HSV color space first. We've written a python script to find the range by adjusting the minimum and maximum value of hue, saturation, and value(HSV). Figure 20 shows a screenshot from the HSV range detection process.

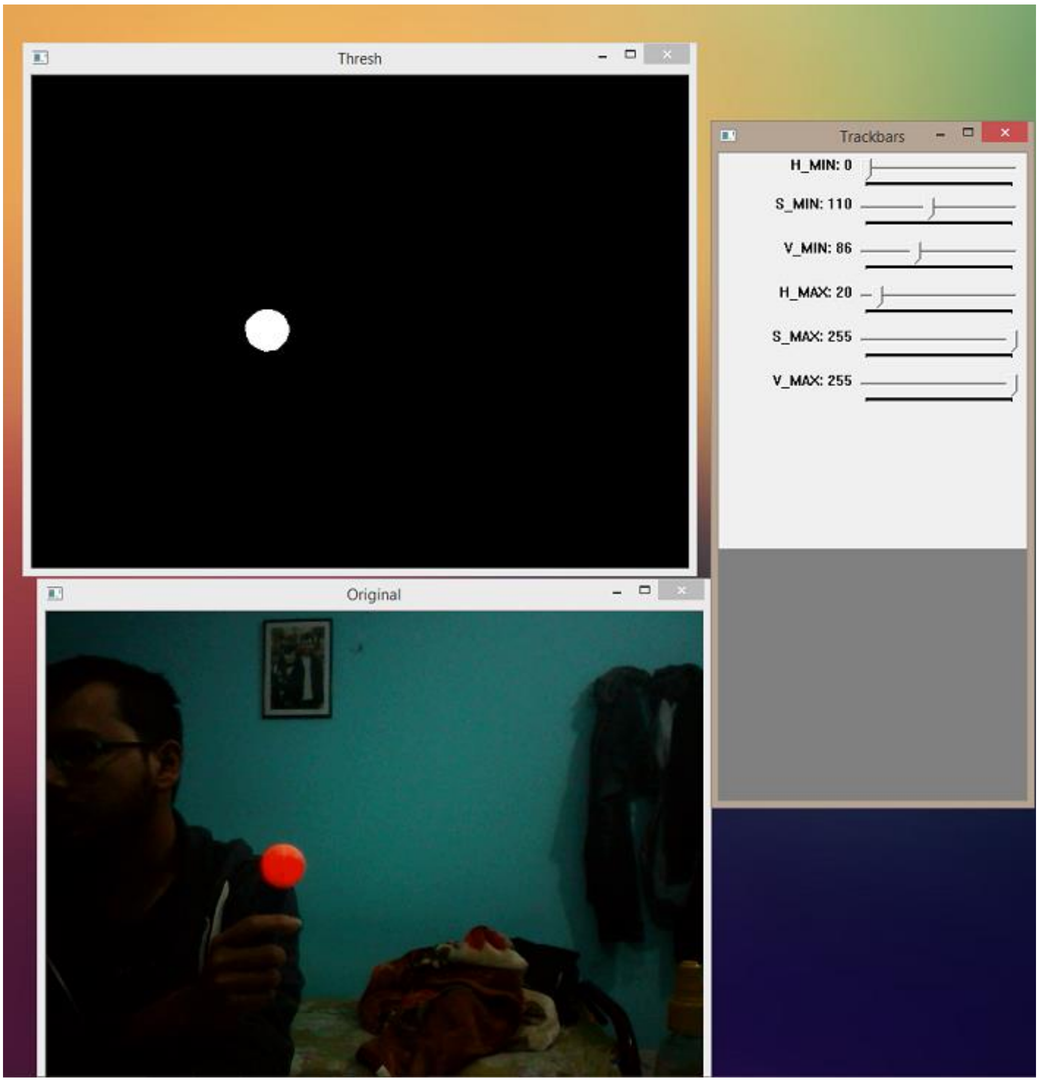

Figure 20: HSV Range Detection Process

3.3.2 Determining Orb Position and Radius from Image

The webcam captures images of the controller and sends it to the processing unit. The light orb on top of the controller is then detected and farther analyzed to find its 3D position. Several image analysis techniques are applied to the captured images to detect the orb. First it is resized to a smaller resolution image for faster processing. Image analysis techniques work better on images in HSV color space than RGB color space. So the captured image is converted from RGB to HSV color space. Then some blurring is applied to the image. This makes the image smoother and reduces noise.

After that the threshold that was determined by Range-Detector script is applied. This converts the color image into a binary image. In an ideal situation, the binary image will have only the orb as a white blob and the rest of the image as black area. But in real world scenarios, there is also a fair amount of unwanted noise that stays after converting into the binary image. To farther reduce the noise, two morphological operation named erosion and dilation are applied to the binary image.

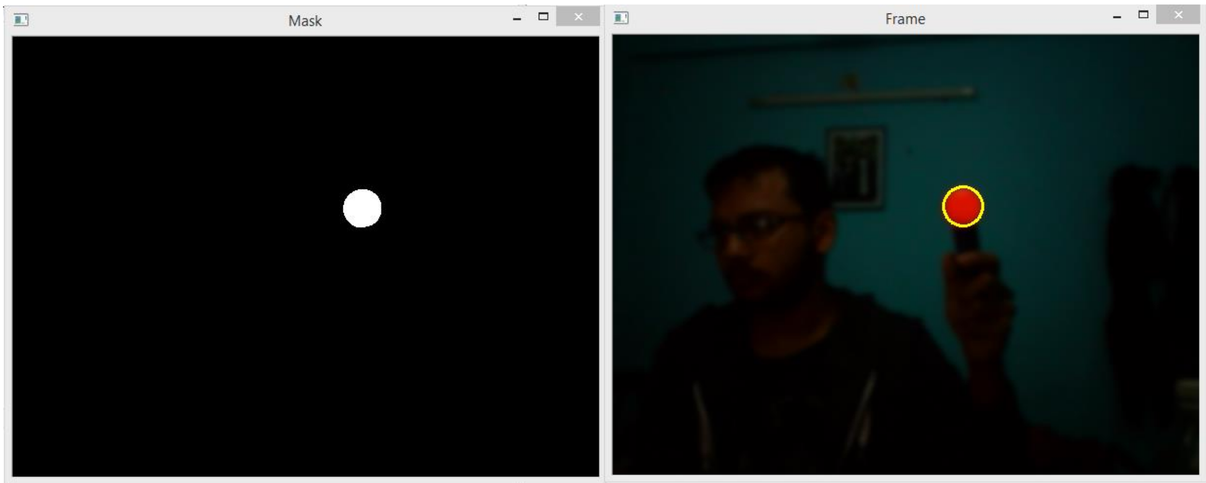

Figure 21: Screenshot from command detection process

Erosion

The basic idea in binary morphology is to probe an image with a simple, pre-defined shape, drawing conclusions on how this shape fits or misses the shapes in the image. This simple "probe" is called structuring element and is itself a binary image (i.e., a subset of the space or grid). [20]

Let **E** be a Euclidean space or an integer grid, and **A** a binary image in **E**. The **erosion** of the binary image **A** by the structuring element **B** is defined by:

$$A \ominus B = \{z \in E | B_z \subseteq A\} \quad \text{......................................(1)}$$

where $\mathbf{B_z}$ is the translation of **B** by the vector ***z***, i.e.,

$$B_z = \{b + z | b \in B\}, \ \forall z \in E \quad \text{.....................................(2)}$$

When the structuring element **B** has a center (e.g., a disk or a square), and this center is located on the origin of **E**, then the erosion of **A** by **B** can be understood as the locus of points reached by the center of **B** when **B** moves inside **A**. For example, the erosion of a square of side 10, centered at the origin, by a disc of radius 2, also centered at the origin, is a square of side 6 centered at the origin.

The erosion of **A** by **B** is also given by the expression:

$$A \ominus B = \bigcap_{b \in B} A_{-b} \quad \text{..(3)}$$

Dilation

A binary image is viewed in mathematical morphology as a subset of a Euclidean space $\mathbf{R^d}$ or the integer grid $\mathbf{Z^d}$, for some dimension **d**. Let **E** be a Euclidean space or an integer grid, **A** a binary image in **E**, and **B** a structuring element regarded as a subset of $\mathbf{R^d}$.
The dilation of A by B is defined by:

$$A \oplus B = \bigcup_{b \in B} A_b, \quad \text{...(4)}$$

where $\mathbf{A_b}$ is the translation of **A** by **b**. [21]

Applying Mask

After performing a series of erosion and dilation process, a box masked is applied to the binary image. This mask is applied to reduce the search space for the light orb's projection in the binary image. More specifically, the orb is only searched in the region of the image where it was found in the previous frame. The region is a square shaped area.

Finding Contours

The binary image is now searched for contours. OpenCV provides a convenient function for that operation. If at least one contour was found in the image, it can be farther analyzed whether it's the orb's projection or not. If a contour is circular with a minimum radius, it is most likely the projection of the orb. In our case, we set the minimum radius of the projected orb to 10 pixels. Since the controller is to be used in a low proximity to the webcam, contours having radius lower than this can be discarded as noise. To test if a contour is round; the ratio of the contours area to the area of the circle that completely encloses the contour can be taken:

$$roundness = \frac{area\ of\ contour}{area\ of\ circle\ with\ radius\ of\ the\ enclosing\ circle\ of\ the\ contour} \quad \text{...(5)}$$

The value of this ratio for a round projection should be 1. But there are some problems with that assumption. Firstly, there is some lens distortion to the projected image depending on the field of view of the camera being used. Most consumer webcams have a limited field of view and so the distortion caused by them are mostly negligible. There is also the problem that the orb's projection is not always retained perfectly after applying the threshold. And finally, when the orb is only partially visible due to occlusion or because portion of it being outside the image frame, the projection will have a contour area to actual area ratio of only 0.5 at max. This diminishes the benefit of using the aforementioned ratio as a criterion for testing roundness. But using it still has a positive effect. When the orb is going out of the camera's view, we only want to detect it up until at least half of it is inside the frame. Anything outside that will create a contour in the image that has an enclosing circle with a radius less than the radius of the full projection of the orb. Using 0.5 as the minimum ratio makes it easy to discard contours once more than half of the orb is outside the image frame. This along with an appropriately chosen radius and the masks applied previously can detect the correct contour in most cases.

The steps of the image based detection process are shown in figure 22.

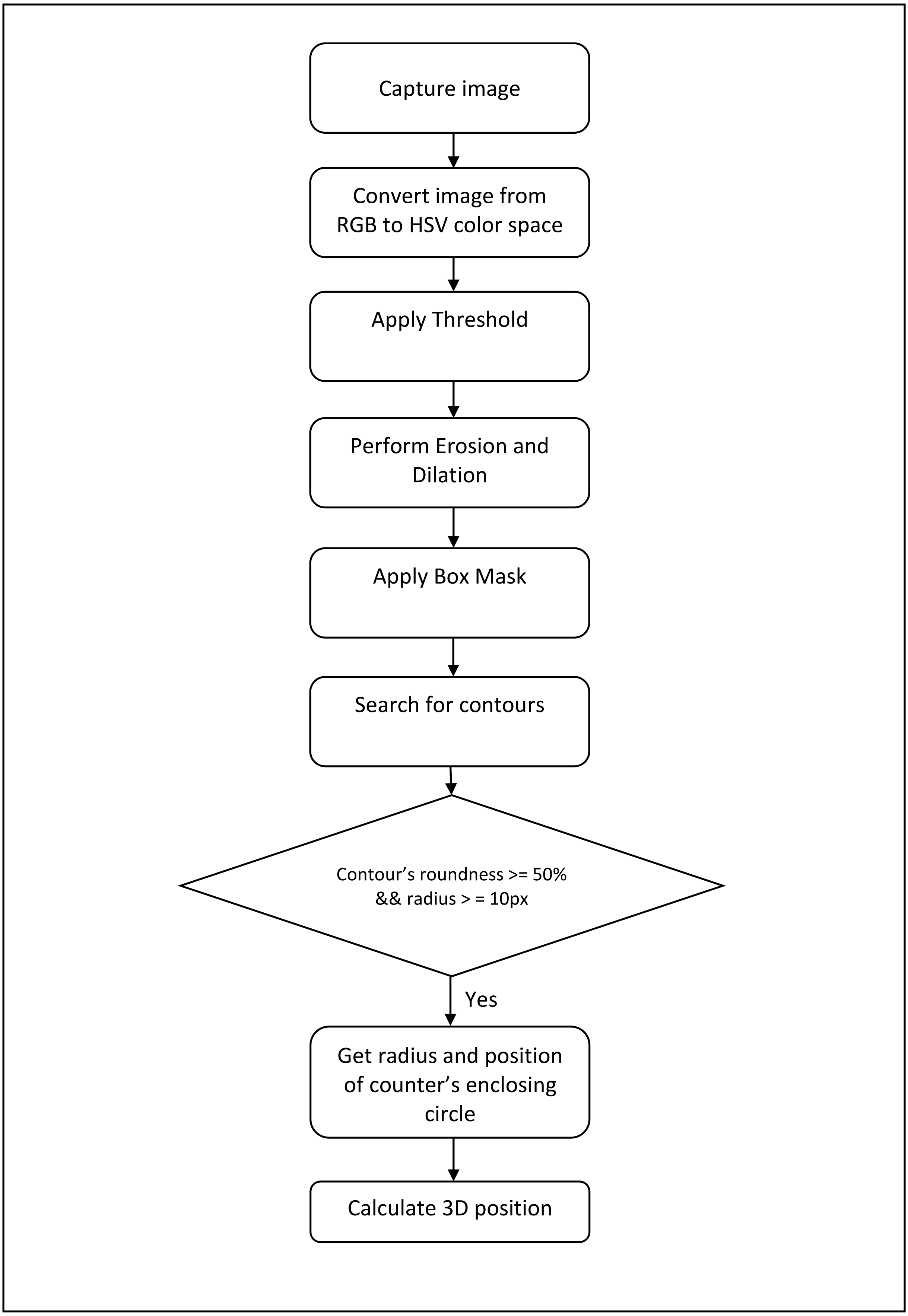

Figure 22: Steps of the command detection process

3.3.3 Determining 3D Position

Once a contour is chosen to be the projection of the orb, the centroid and the radius of the enclosing circle is calculated. They are then used together to calculate the 3D position of the orb and hence the users hand holding the controller. Suppose, the orb is at P(x,y,z) position in space. Then the value of z will be,

$$z = \frac{W.F}{P} \qquad \text{...(6)}$$

Where,

z = Distance from camera

W = Known width of the orb

P = Measured radius in pixels

F = Focal Length

The, focal length is an intrinsic property of the camera being used and varies depending on the camera being used. It is calculated as:

$$F = \frac{P.D}{W} \qquad \text{...(7)}$$

Where,

D = The known distance of the orb from the camera when measuring F

P = Radius of the orbs projection in pixels

Now using the z value just calculated, the value of x and y will be,

$$x = \frac{z.(u - p_x)}{F} \qquad \text{..(8)}$$

$$y = \frac{z \cdot (v - p_y)}{F} \quad \text{...(9)}$$

Where,

(u, v) = Position of the center of the orb's projection in image

(p_x, p_y) = Center co-ordinate of the image frame [22]

3.3.4 Command Detection

The interaction controller also allows for user commands by blinking of the light orb. Once a blink of the orb is detected it is interpreted as a click event. If the detection of the light orb was lost for a fraction of a second, for example 500 millisecond; it would be detected as a click event.

Once a new position is calculated, it is pushed on to a FIFO queue. One of the challenges in calculating the positional data from the image is that, unless the detection is very accurate, the calculated 3D position is not stable. The positional values tend to fluctuate even when the controller is being held steadily. To reduce these unwanted fluctuations, we've used moving average to the data. Although Simple Moving Average (SMA) algorithm smoothens the data, it has a delay associated with it that makes it less responsive to changes in direction. To reduce this lag and increase responsiveness, another moving average technique, the Hull Moving Average (HMA) [23] algorithm was applied. This technique not only smoothens the data but also reduces lag. HMA needs a history of previous positional data to calculate the new position. That is why we keep a history of the most recent positions in a queue.

3.4 Transmission of Motion Data

The positional data and click events, if any, are sent to the application running on the smartphone device wirelessly. We have used UDP for sending out packets of data. UDP was chosen because we don't need the reliability of TCP and because we need the fastest possible transfer rate. In fact, we only require the latest positional data. Every packet is stamped with a sequence number. Using this number VR application can take into account only the packets with the highest sequence numbers and discard packets that have arrived late. The packets are sent using a different thread other than the image analysis thread to quicken the sending of packets.

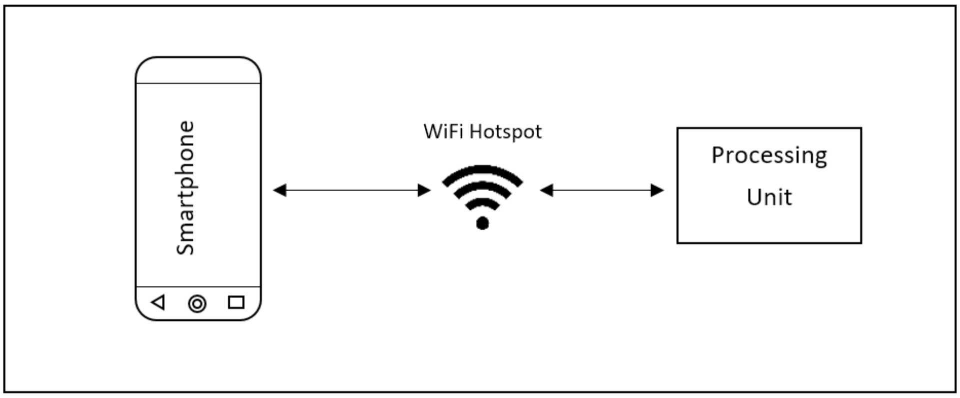

Figure 23: Data transmission using a hotspot

3.5 VR Application

The smartphone application is a structure building and VR painting application. The application allows the user to build structures using blocks of brick. The controller is used to select, move and place a block in the desired position. The application receives the command that was sent from the processing unit wirelessly. It then acts according to those commands. The command can be a position of the orb or a click event. The click event is determined from a blink of light orb. A click event is used to grab a block, while next click releases a block and place it in the position where it was released. In paint mode, the controller acts as the paint brush of the user. The user can select from different colors and paint in a canvas. In figure 24, we've shown a screenshot from structure building mode. Implementation details of the VR application are described in the next chapter.

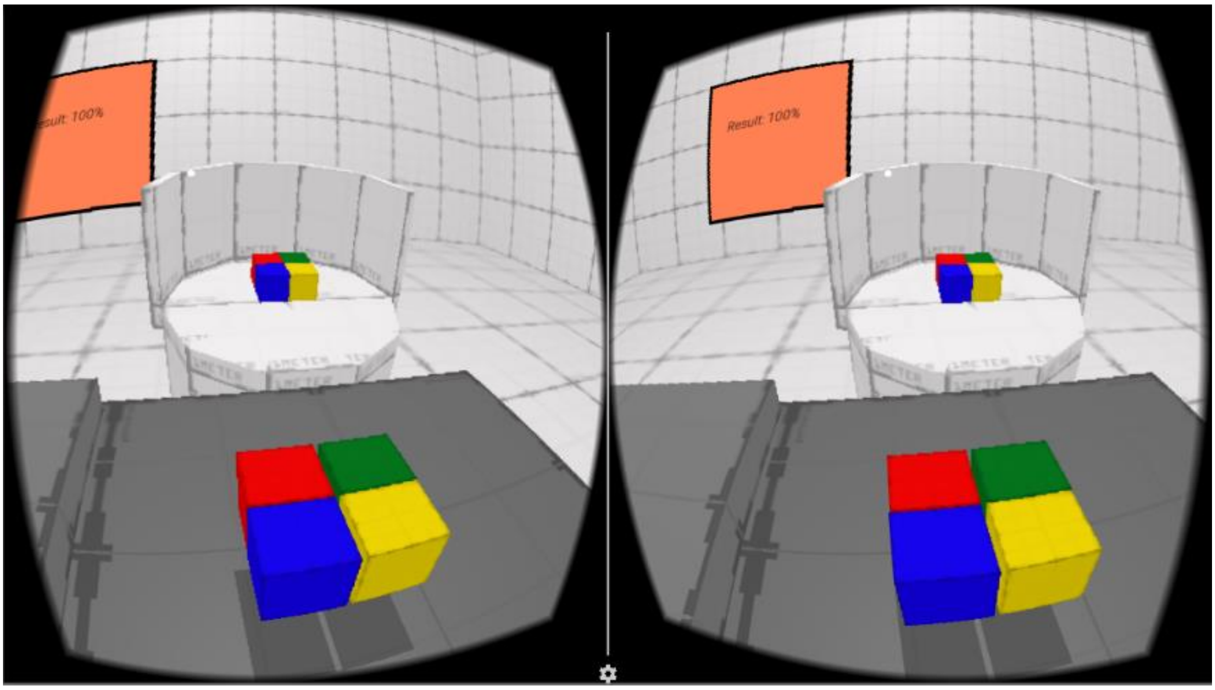

Figure 24: An in-game screenshot

CHAPTER 4

Edutainment Application Development

4.1 Tools & Technologies

Among the two main parts of our system, the controller doesn't require any software. In the edutainment application part, we've used **Unity Engine**. Scripting is done in **C#** language. We've used a PC powered by windows 8 as the processing unit. For controller detection from image, **OpenCV** is used. For server configuration and transmission, we've used **python** socket. The following figure, figure 25, is the diagram showing the tools used in the application and the server.

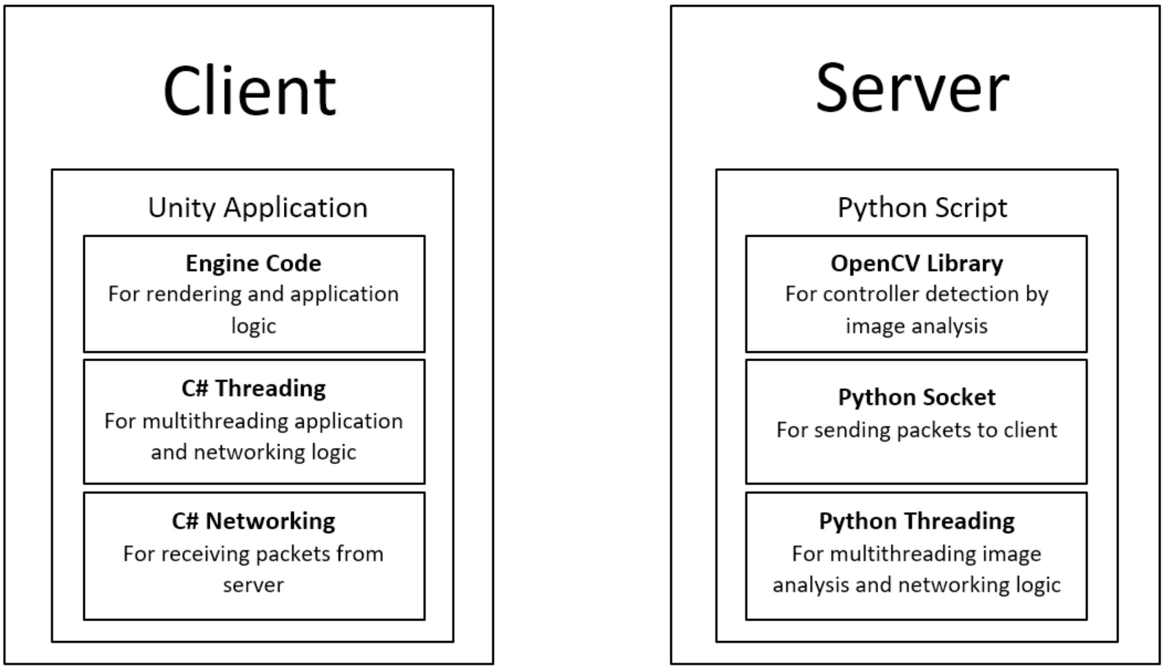

Figure 25: Tools and technologies used in client application and server

4.2 Edutainment Game

The edutainment application we have developed has three modes. Two of the modes are concerned with arranging blocks and making 3D structures. The third mode is a VR painting application. The game features and interaction techniques are described next.

4.2.1 Game Features

The three modes of the application are:

1. Sandbox Mode,
2. Challenge Mode,
3. VR Paint Mode.

We're going to describe these three modes next.

1. Sandbox mode:

In this mode the user is given a collection of colored blocks. It is an open ended game mode with no definitive goal. The user can pick up and arrange the blocks in any way he/she wants. There is also an unlimited supply of blocks so that once the user picks up a block, a new block is spawned in its position.

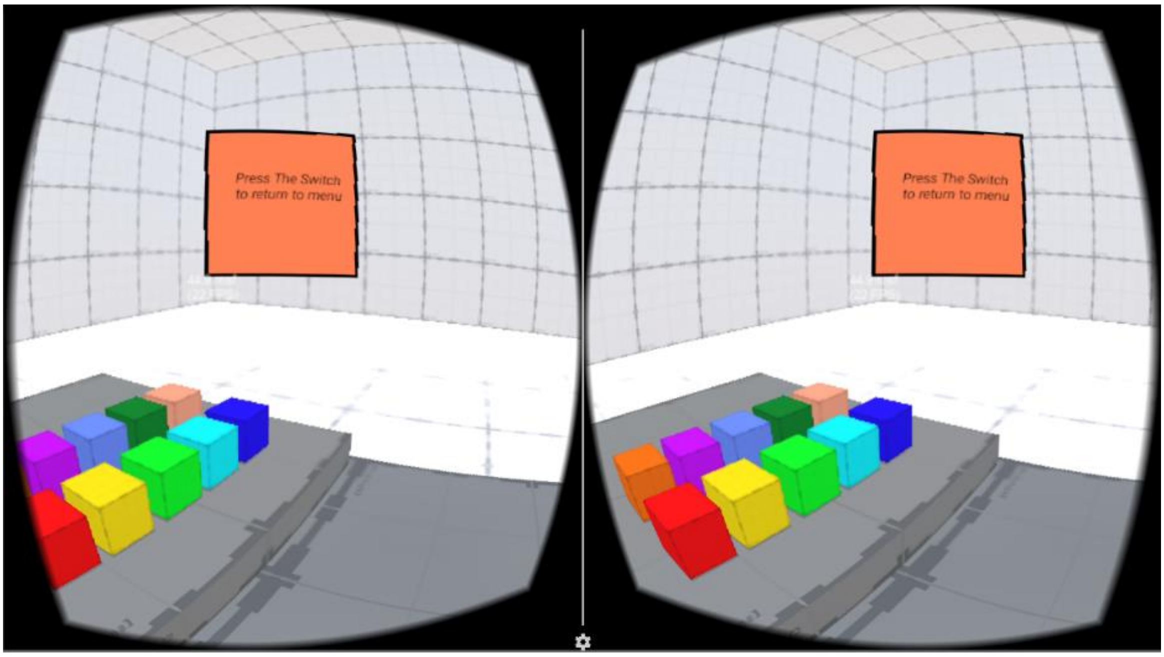

Figure 26: On sandbox mode, anything can be built

2. Challenge mode:

This mode tasks the user with arranging the blocks to match a design. The target design is first shown to the user and then hidden. The blocks in the target design are disassembled and put in front of the user. The user then tries to arrange these blocks correctly. The challenge mode works as a type of memory game.

3. VR Painting Mode:

The user is given a canvas to paint on. He/she is also given a set of colors to choose from. The user moves a paint brush. Putting and dragging the paint brush on the canvas puts color on the canvas. The user can change the currently selected color by putting the paint brush in the desired color palette.

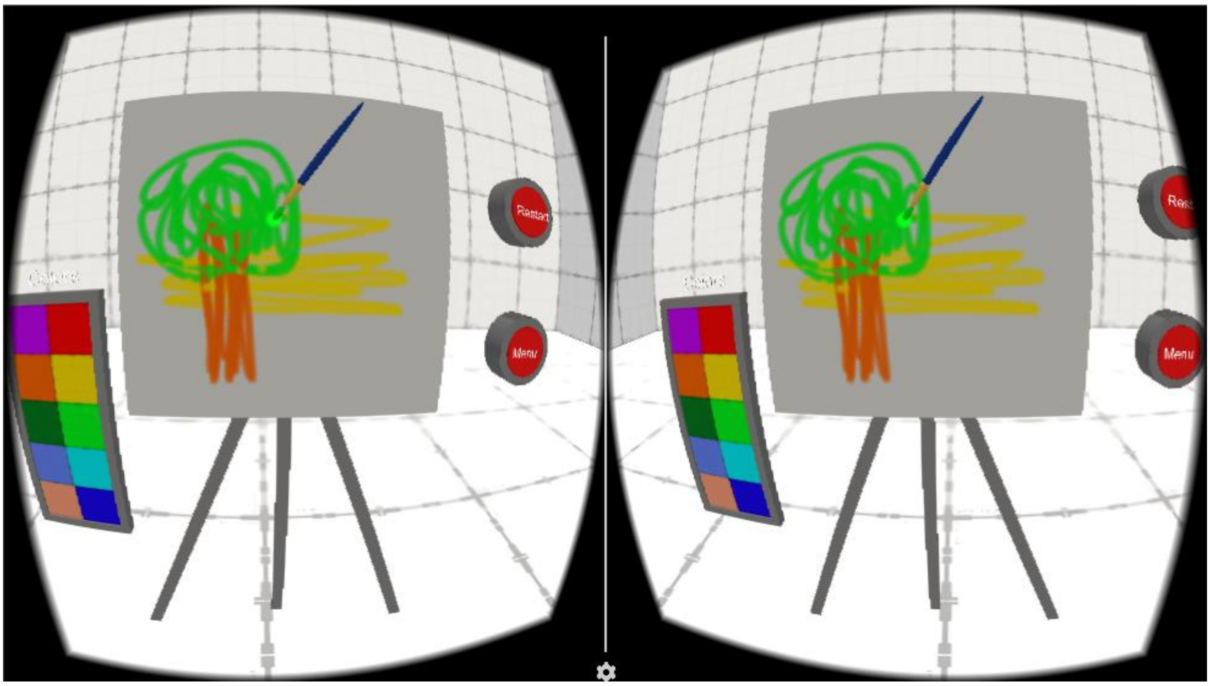

Figure 27:The controller acts as a paint brush in VR painting mode

4.2.2 Game Interaction

General interaction in both sandbox and challenge mode is the same. The position of the controller in the virtual world is represented as a white marker. Moving the controller in the real world moves the marker in the virtual world. Click events of the controller are used to select and deselect blocks as well as push switches.

The painting mode is different from these two modes. Here the motion of the controller is translated as the motion of a paint brush. The click event wasn't used in this mode except for interacting with virtual switches.

4.2.3 Implementation Details

In the blocks modes, the user selects a block by first putting the marker near the block followed by a click. Once a block is selected, moving the controller moves the block with it as well. Another click causes the block to be deselected. So the block doesn't follow the controller anymore.

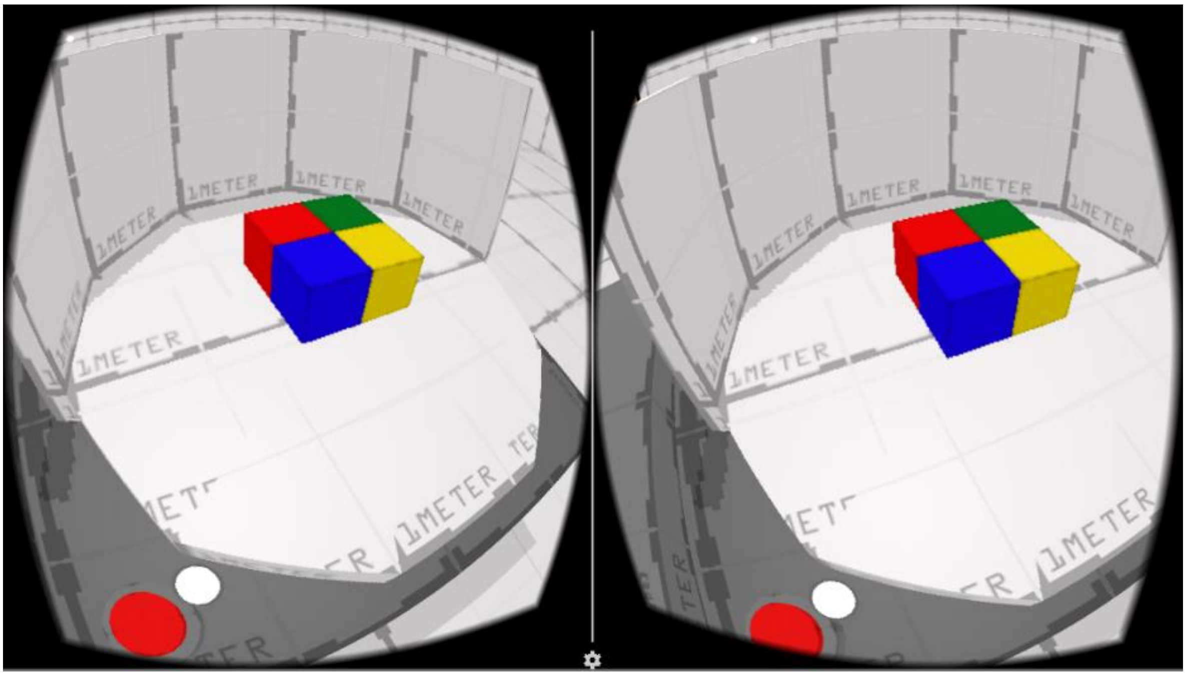

Figure 28: The white marker near the red start switch

If a selected block is moved near to a resting block, the resting block turns white with a black outline. This is to indicate that if the user releases the selected block now, the selected block will automatically place itself properly by the closest face of the resting block. This makes it easier to place the blocks just beside other blocks. Ray casting from all sides of the selected block was used to find if there is any other nearby block.

In the challenge mode, we needed to test if two structures have the same arrangement. The two structures being the original structure that the user is trying to recreate and the user created structure. We devised an algorithm for doing this. Each block was represented with a block data structure. Each blocks data structure contained a list of the blocks in its up, down, left, right, front and back positions. Whether there's a block in any of those positions is found by casting short length rays in each of those directions. If a ray hits a block, then that information is stored. This is done for all the blocks in both of the structures. After that six 2D arrays, each for one of the six sides of a block, are initialized to 0. The arrays have rows and columns equal to that of the number of different colors for blocks. We have used blocks of 10 different, so the arrays are 10 by 10 in size.

We now go through each block in the original structure. If, for a block of color ‘a’ there is an adjacent block of color ‘b’ on the side ‘s’, we increment the value at [a,b] of the array for side ‘s’. Here ‘a’ and ‘b’ are integer values between 0-10 used to represent each of the colors.

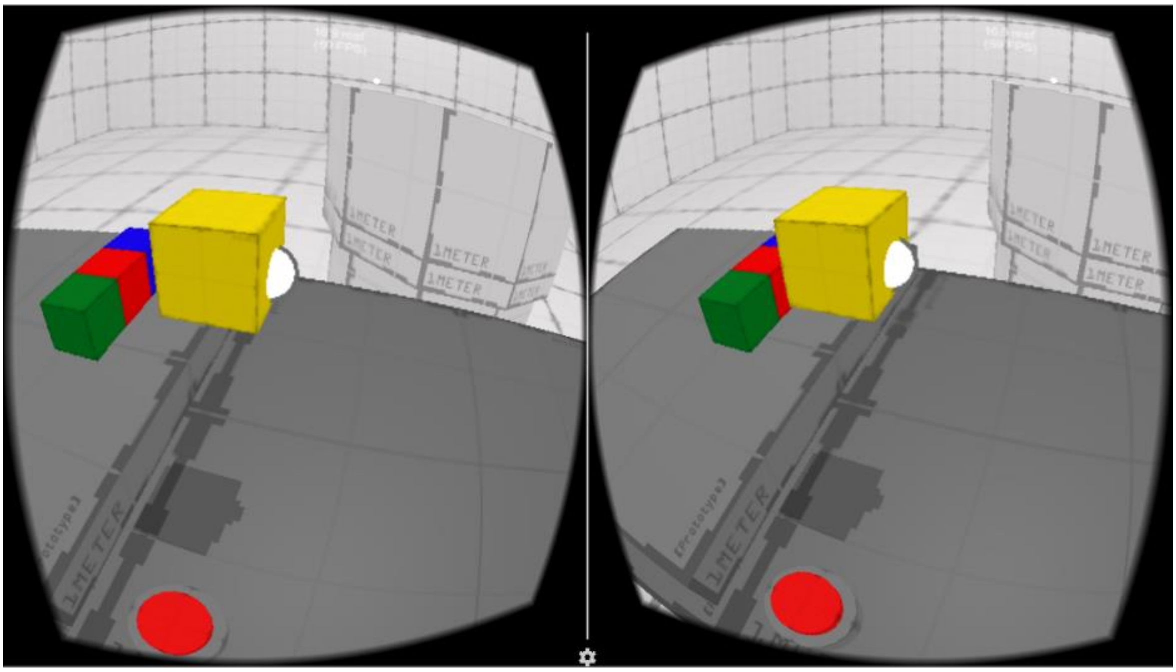

Figure 29: Selecting and moving a block using the controller

Once this is done, the same thing is done for the user arranged structure. But instead of incrementing the value at [a,b], it is now decremented.

Finally, we check all 6 arrays to find any non-zero values. If a non-zero value is found, that would mean the two structures didn’t match completely. Here, we couldn’t have done this with a single array instead of six arrays. Because two blocks can be adjacent in six different ways. That is why knowing only if two blocks are adjacent or not is not enough to know exactly how they are arranged.

We can summarize the above steps in brief as follows:

---

1. *get the list of every block adjacent to every other block in original structure*
2. *get the list of every block adjacent to every other block in user created structure*
3. *initialize six 2D arrays to 0 each for the six sided of a block*
4. *For any block 'a' and 'b' in original structure; if' 'b' is adjacent to 'a' at side 's' of 'a', increment s[a,b]*
5. *For all blocks 'a' and 'b' in user structure; if 'b' is adjacent to 'a' at side 's' of 'a', decrement s[a,b]*
6. *for any s[a,b] , if s[a,b] is non-zero the two designs don't match*
7. *otherwise match was found*

In the painting mode, the user paints on a virtual painting canvas. Although it is much easier to directly manipulate the color of the canvas' texture, it is not suitable for real-time application usage. It takes too long to update texture data and is unusable for such application. So instead, Unity's Sprite GameObjects are instantiated whenever the user paints on the canvas. Each Sprite is basically the image of a filled circle. Consecutive Sprites placed at close proximity in a line gives the impression of a brush stroke.

The Sprites are not instantiated on the canvas directly. In fact, they are instantiated outside of the user's view. An orthographic camera captures the image of this and renders it to a RenderTexture object. This RenderTexture is used as the texture for the canvas. Since we are instantiating the Sprites very frequently, to reduce unnecessary garbage collection, a large pool of Sprites is instantiated in the beginning and reused.

4.3 Limitations

Image processing in our system requires a PC with a webcam. We've determined the range of HSV manually. As the detection process is dependent upon colors found in the image, this won't work perfectly in brightly or unevenly lit environment.

CHAPTER 5

Conclusion and Future Works

5.1 Conclusion

Virtual reality is a technology where the user can immerse himself fully in an imaginary world. Edutainment is an idea of educating people subconsciously through entertainment. These two have a common factor to utilize that is sub-consciousness of the user. So VR provides a more effective way of edutainment. This is why we've done this research to take the VR edutainment experience one step further by using an interaction controller that doesn't require any wired or wireless connection with the system. The detection is based on image analysis. We've developed a demo application to experiment the effectiveness of VR with the controller. The controller plays the most important part as the medium of interaction in the VR world. This idea can be used to build many more applications like pilot training in a simulated environment, educating children while playing games, sports training, making architectural structures in the 3D world, 3D paint application and many more.

5.2 Future Works

We've developed this application using a PC for image-based analysis. With more advancement in smartphones, it might be possible to do the entire processing in the background while running the VR application in the foreground of the device. This would eliminate the necessity of the external camera by using the smartphone camera for detecting the controller.

Bibliography

[1] "What is Virtual Reality? - Virtual Reality," Virtual Reality Society, 2016. [Online]. Available: http://www.vrs.org.uk/virtual-reality/what-is-virtual-reality.html. [Accessed 30 July 2016].

[2] "Educational entertainment - Wikipedia," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Educational_entertainment. [Accessed 17 January 2017].

[3] C. Payatagool, "Theory and Research in HCI: Morton Heilig, Pioneer in Virtual Reality Research - Telepresense Options," Telepresence Options, 19 September 2008. [Online]. Available: http://www.telepresenceoptions.com/2008/09/theory_and_research_in_hci_mor/. [Accessed 30 July 2016].

[4] I. E. Sutherland, "The Ultimate Display," in *Proceedings of IFIP Congress, pp 506-508*, 1965.

[5] K. Horowitz, "Sega VR: Great Idea or Wishful Thinking?," Sega, 28 December 2004. [Online]. Available: http://www.sega-16.com/2004/12/sega-vr-great-idea-or-wishful-thinking/. [Accessed 30 July 2016].

[6] "The Oregon Trail (video game) - Wikipedia," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/The_Oregon_Trail_(video_game). [Accessed 20 January 2017].

[7] "Odell Lake (video game) - Wikipedia," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Odell_Lake_(video_game). [Accessed 19 January 2017].

[8] C. Anthes, R. J. García-Hernández, M. Wiedemann and D. Kranzlmüller, "State of the Art of Virtual Reality Technologies," in *2016 IEEE Aerospace Conference*, Big Sky, Montana, United States, March 2016.

[9] D. Nield, "How Oculus Rift works: Everything you need to know about the VR sensation," Wareable, 29 March 2016. [Online]. Available: http://www.wareable.com/oculus-rift/how-oculus-rift-works. [Accessed 30 July 2016].

[10] A. Robertson, "The ultimate VR headset buyer's guide," The Verge, [Online]. Available: http://www.theverge.com/a/best-vr-headset-oculus-rift-samsung-gear-htc-vive-virtual-reality#comparisontable. [Accessed 21 January 2017].

[11] "HTC Vive - Wikipedia," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/HTC_Vive. [Accessed 22 January 2017].

[12] "PlayStation VR - Wikipedia," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/PlayStation_VR. [Accessed 24 January 2017].

[13] "Google Cardboard - Wikipedia," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Google_Cardboard. [Accessed 24 January 2017].

[14] "Samsung Gear VR - Wikipedia," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Samsung_Gear_VR. [Accessed 24 January 2017].

[15] "New high-performance cave automatic virtual environment that functions with gesture recognition," Phys.org, 18 April 2013. [Online]. Available: http://phys.org/news/2013-04-high-performance-cave-automatic-virtual-environment.html. [Accessed 3 August 2016].

[16] Y. A. G. V. Boas, "Overview of Virtual Reality Technologies," in *Interactive Multimedia Conference 2013*, 2013.

[17] T. Schlomer, B. Poppinga, N. Henze and S. Boll, "Gesture Recognition with a Wii Controller," in *2nd international conference on Tangible and embedded interaction*, Bonn, Germany, 2008.

[18] "Google Arts & Culture VR - Android Apps on Google Play," Google, [Online]. Available: https://play.google.com/store/apps/details?id=com.google.vr.museums&hl=en. [Accessed 21 January 2017].

[19] "Expeditions - Android Apps on Google Play," Google, [Online]. Available: https://play.google.com/store/apps/details?id=com.google.vr.expeditions. [Accessed 21 January 2017].

[20] "Erosion (morphology) - Wikipedia," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Erosion_(morphology). [Accessed 23 January 2017].

[21] "Dilation (morphology) - Wikipedia," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Dilation_(morphology). [Accessed 23 January 2017].

[22] D. Bradley and G. Roth, "Natural interaction with virtual objects using vision-based six DOF sphere tracking.," in *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology (pp. 19-26). ACM.*, 2005.

[23] A. Hull, "How to reduce lag in a moving average," [Online]. Available: http://www.alanhull.com/hull-moving-average. [Accessed 25 January 2017].

[24] S. P. A. W. a. E. W. R. Fisher, "Glossary - Binary Images," Informatics Homepages Server, University of Edinburgh, [Online]. Available: http://homepages.inf.ed.ac.uk/rbf/HIPR2/binimage.htm. [Accessed 30 July 2016].

[25] "Likert scale - Wikipedia," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Likert_scale. [Accessed 19 January 2017].

[26] "How to Find if Triangles are Similar," MathsIsFun.com, [Online]. Available: https://www.mathsisfun.com/geometry/triangles-similar-finding.html. [Accessed 19 January 2017].

[27] D. Nield, "How Oculus Rift works: Everything you need to know about the VR sensation," Wareable, [Online]. Available: https://www.wareable.com/oculus-rift/how-oculus-rift-works. [Accessed 23 January 2017].

[28] P. Lamkin, "Sony PlayStation VR: Essential guide to the hardware, games, release and more," Wareable, [Online]. Available: https://www.wareable.com/project-

morpheus/sony-project-morpheus-release-date-price-games. [Accessed 23 January 2017].