PROJECT REPORT

SRI LANKA TECHNOLOGICAL CAMPUS

VR STIMULATIONS

NAME & REG NUMBER: AKEEL MOHAMAD – BSC/EE/2016/01/0001

SANJUKA KARUNATHILAKA – BSC/EE/2016/01/0002

VINURA KARUNARATNE – BSC/EE/2016/01/0007

CHAMATHKA LAKSAHANI – BSC/EE/2016/01/0013

AMALI RAJAKARUNA – BSC/EE/2016/01/0015

HESHAN NIRANGA – BSC/EE/2016/01/0023

SHANITH GAMAETHIGE – BSC/EE/2016/01/0028

SUPERVISOR'S NAME: MR. VIRAJ SAHABANDU

DATE : 20/02/2018

DECLARATION

We, Group J , hereby declare that the project titled " VR Stimulations " was presented by us under the supervision of Mr.Viraj Sahabandu.

This project work, under the Module Code 2301, is submitted by us in the partial fulfilment of the requirements of Bachelor of Science Honors in Engineering study program of the Sri Lanka Technological Campus. We further declare that this submission is a result of our own work, except as cited under reference or quoted with acknowledgements, and has not been submitted to any other University or Institution in the candidature of any other degree or an equivalent qualification.

Date: 20/02/2018

"I certify that I supervised the project of Group J, VR Stimulations and in my opinion this report is sufficient in terms of scope and quality of undergraduate level."

Name: Mr. Viraj Sahabandu

Signature: Date: 20/02/2018

CONTENTS

ACKNOWLEDGEMENT

The success and the final outcome of this project required a lot of guidance and assistance from many people. We would like to express our deepest appreciation to all those who provide us the possibility to complete this project.

A special gratitude we give to our supervisor Mr. Viraj Sahabandu for his Guidance and Constant supervision as well as providing necessary information regarding the project. And also we wish to express sincere gratitude to the staff of the Sri Lanka Technological Campus for the guidance and support.

We are thankful to Mr.Udesh Oruthota, Senior Lecturer, Sri Lanka Technological Campus and Mrs.Chandani Dissnayake, Senior Lecturer, Sri Lanka Technological Campus.

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.

We would like to express our gratitude towards our parents for their kind cooperation and encouragement which help us in completion of this project.

ABSTRACT

Immersive Human Machine Interaction (Immersive HMI) is the next generation human-machine user interface that all computer scientists & engineers are eyeing on. Through the usage of modern technologies such as Indoor localization, game engine simulation & real time IMU sensing we tried our best to achieve next generation virtual reality simulation that will serve a foundation for the further enhancement for more industry standard military simulations that we are planning to enhance in future.

CHAPTER 01 – INTRODUCTION

This project report documents the analysis, define and development of the VR Stimulation project, and documents thoroughly different work and analysis within the relevant space.

1.1 Project Outline

This project, "VR Simulation" developed with the intention of giving an immersive visual output to the user who is moving & rotating in the real world in order to interact with the virtual scene. This prototype contains and portrays virtual scenes and objects, combined with an experimental interaction techniques and physical simulation within a video game based system. This entails the use of experimental re-appropriated technologies in both the creation of, and interaction with virtual reality content.

1.2 Motivation

With the increasing processing power and the increased usage of the smart devices like mobile phones, laptops & handheld gaming consoles, virtual reality technology is becoming more accessible to the public, and even entry-level computers are capable of rendering more complex real time augmented reality scenes. By replacing traditional symbolic user interaction techniques with more natural methods, the interface can be designed to be more intuitive, as it returns the semantics of motion and interaction to that which is more instinctive to the user. As the new generation is increasingly situated within the virtual world, rather focusing on more complex & powerful user applications, we should also develop new user interaction techniques that is more intuitive to user like haptic feedback & Virtual reality video output than the traditional video screen visual output.

As the video game & multimedia industry continues to grow, game & graphics developers allocating a reasonable amount of time & resources to focus on new user interaction mechanisms of relating haptic & visual feedback to the user. Specially this VR simulation becoming more popular towards mimicking some harsh, difficult environments virtually within a single indoor room, that will be more useful to simulate military training, submarine, aircraft & space simulations for beginners and amateurs In related field.

There is a potential for new user interaction techniques as researches are focusing on even more intuitive user interactions like "Neurable's" mind controlled VR headband. After the development of more advanced & powerful computing devices now there's a rapid blooming towards the development of new user interaction techniques which will replace the traditional user interaction techniques like keyboard, mouse, joystick, monitor to more intuitive physical interactions & haptic VR visual outputs.

1.3. Objectives

The main aim of our project is to create a Virtual reality system by the use of variety of hardware & software tools to capture the motion of the user and output the related visuals to his/her respective position. That will serve as an art-based representation of the increasing fusion of physical and virtual data, in terms of the information that people communicate, the culture that they create and spread, and the lives that people live.

1.4 Structure

This project is comprised of three main classifications,

1. Capturing the orientation of the view point of the user in 3D space & his relative Cartesian position from the initial point where the simulation begins, by the use of motion sensors (6 DOF)

2. Interpreting the raw data from the sensors in real time in order to obtain more accurate information about user's viewpoint orientation and his relative position.

3. Output the VR scene to the user in real time accordance with his/her current position & orientation.

For the sake of this project, the VR scene refers to the visual output delivered in the software (unity game engine) that run the game. By the use of standalone VR headset, (Oculus Rift preferred), we can simulate those VR scenes directly to user's eye point in way that is more intuitive.

1.5 Installation

The prototype is best suited to simulate a graphical scene within a software, based on the orientation & position of our motion tracking sensors, that could be install in a user's VR Headband, so that the system will track the position & the orientation of the user's eye. This prototype is currently just captures the 3D position & 3D orientation (6DOF) of our sensor unit and simulate the VR scene in the unity engine based on this data. However, by combining this sensor unit with a VR headset & outputting the unity scene to that VR headset, we could make this simulation more elaborately & can use for military, aircraft simulations too.

CHAPTER 02 – RESEARCH

For this project, research was done into a variety of different areas that would be of most advance to the development of the project. Particularly, the research areas are indoor localization techniques, trilateration algorithms & implementation, indoor positioning using IMU sensors, kalman & complimentary filtering & game engine technology.

2.1 Technology Research

These section frameworks the details of our selection of technologies that we researched and contains relevant theories & concepts.

2.1.1 Indoor Localization Techniques

An **indoor positioning system (IPS)** is a system to pin point objects or people inside a building consuming radio waves, magnetic fields, acoustic signals, or additional intellectual information assembled by mobile devices. There are numerous viable systems on the market, but there is no standard for an IPS system.

Knowing the location & determining the location are the essential part of our projects as those are the inputs for our VR stimulation. In contrast to GPS (Global Positioning System) IPS'es use various techniques to determine position including, distance measurement to nearby anchor or beacon nodes (nodes with fixed or known position e.g. Wi-Fi access points, Bluetooth beacon, fixed radio or ultrasonic transmitter), magnetic positioning & death reckoning.

Since our VR stimulation placed in an indoor location and it demands the accuracy of our location measuring up to a sub centimeter accuracy, we cannot go for GPS positioning as it uses TOF (Time of Flight) measurements from the geo stationary satellites which prone up to 20 meters of less accuracy which can't be tolerable for our application. Therefore, for this project we should opt for a much accurate small-scale indoor positioning system in order to track the position & the orientation of the users. There are various techniques such as RF and ultrasonic transmitter receiver trilateration used in commercial scales but due to its high technical complexity in its installation & the high cost, we opt for a location tracking system using IMU sensors.

2.1.2 Trilateration Technique

The trilateration technique is the most common technique in commercial uses of any indoor localization applications. As it names suggest, there should be at least three fixed nodes with known coordinates which either emits or receive any particular electromagnetic signal, can be used for approximate the location of any mobile node which is situated within those fixed nodes coverage. The trilateration principle can be explained as follows. Assuming that emitters and receivers are synchronized, i.e. their clocks have exactly the same time without significant delays, if at the receiver of unknown position (x,y,z) we measure the TOF, tk, to each satellite k, then multiplying them by the speed of the propagating wave, c, we obtain a set of ranges from the receiver to each individual satellite, rk. Writing down the equation of a sphere of radius rk centered at each satellite at position (xk,yk,zk) (eq. 1), we obtain a system of N equations (N ≥ 3), whose solution tell us the receiver or user position (x,y,z).

$$\sqrt{(x_k - x)^2 + (y_k - y)^2 + (z_k - z)^2} = r_k \qquad \textbf{(Eq. 1)}$$

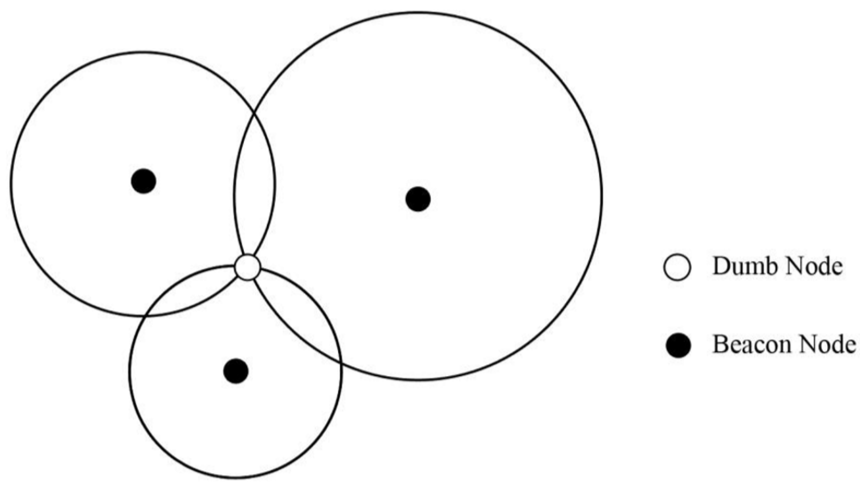

Figure 1

2.1.3. IMU Sensors

IMU (Inertial Measurement Units) are self-contained electromagnetic sensor unit, which measures the change in linear and angular motions usually with a 3-axis accelerometer and the 3-axis gyroscope. IMU’s are typically used for determine orientation & position of any particular device especially in smartphones to measure the tilt for various needs. Almost all the modern smart phones & tablets use this IMU for motion tracking.

For our project, since trilateration technique having more signal processing, syncing & high cost considerations, we opt for IMU for our location tracking purposes. For that case, we used both 3-axis accelerometer & 3-axis gyroscope for determine both position & orientation of our tracking device relative to the initial point by using the technique called death reckoning. Since IMU's only measures change of motion (i.e. physical acceleration along a axis by accelerometer & the angular velocity along a axis by a gyroscope) we have to integrate those values over the time from the initial point of the stimulation in order to determine the position & the angles of the orientation in current time. This process of calculating current position by using previously determined position in navigation is what we call death reckoning.

As we analyze this process, we could clearly see the major flaw of this technique, that is the drift in the final measurement over the time increase with time since this process dynamically depends with the measurement of previous one in order to obtain the current one.

2.1.3.1 Gyroscope

Gyroscope is a spatial mechanism, which processes the rate of change of the angle along a certain axis over a time by measuring the gimbal revolution over the time.

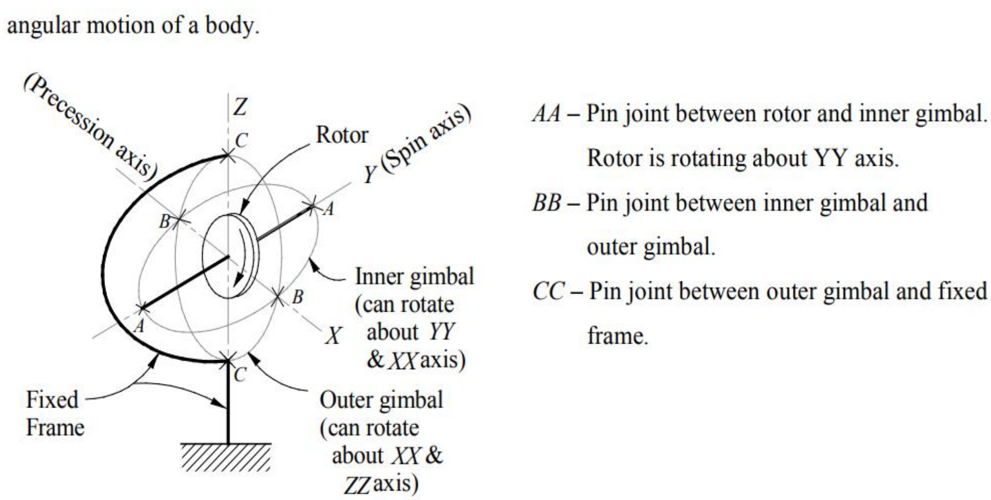

Figure 2 - Gyroscope

Gyroscope can give readings in either analog or digital values. Analog gyros are simply output an analog voltage scaled to either 0V-5V or 0V-3.3V based on the reference voltage of preference with a fixed sensitivity (scale factor).

For example, if a datasheet of an analog 1 axis gyroscope printed as

- Offset (output at angular velocity = 0): **2.65V**
- Sensitivity (scale factor): **107.42 mV/deg./s**

Then the angular velocity can be obtained by

$$\omega = \frac{V_{out} - 2.65}{107.42\ mV/deg./s} \qquad \textbf{(Eq. 2)}$$

Nevertheless, in digital gyroscope, values will be given by a digital value based on a scaling range determined by the datasheet of the sensor. Since our application needs more specific data, we opt for a digital one, which will be discuss within the later half during this report.

2.1.3.2 Accelerometer

Accelerometers is another essential IMU sensor for death reckoning position tracking. It calculates the acceleration towards X, Y, Z axis related to its body frame by measuring physical forces acted along each axis. Since accelerometers not directly measures acceleration and but the force act upon it. Even when the sensor is in rest, there is always a +1g acceleration reading along Downwards accountable for the gravitational force.

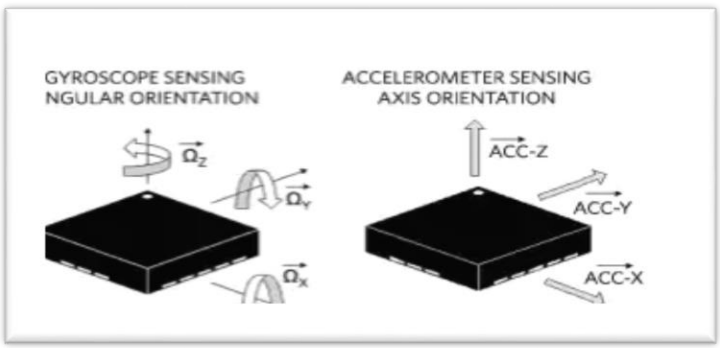

Figure 2.1 – Angular & Linear motion sensing

The Accelerometer's Sensing Mechanism

Most modern day accelerometers sense acceleration by using a mechanism that detects capacitance of a moving mass. A moving electrode mass is attached with a spring, located parallel and opposite to a fixed electrode (as shown in below figure) which is reactive to the physical forces, then by measuring the rate of change of capacitance over the time of that moving electrode, the circuitry will measure the force act along that direction. Once the force is measured, by using Newton's second law acceleration can be calculated.

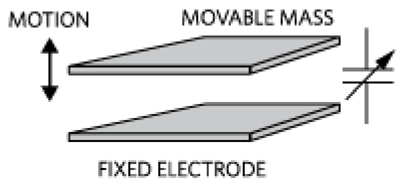

Figure 2.2 - Moving mass and capacitance

This sensing technique is identified for its high precision, stability, low power dissipation, and simple structure to build. It is not prone to noise and change with temperature. Bandwidth for a capacitive accelerometer is only an insufficient hundred Hertz because of their physical geometry and the air stuck inside the IC that appears as a damper.

The capacitance can also be established as single-sided or a differential pair. Let's look at accelerometers arranged as a differential pair (**shown in below figure**). It is composed of a particular movable mass (one planar surface), that is arranged along with a mechanical spring between two, fixed, reference silicon substrates or electrodes (another planar surface). It is obvious that the movement of the mass (Motion x) is qualified to the fixed electrodes (d1 and d2), and affects a change in capacitances (C1 and C2). By analyzing the difference between C2 and C1 we can develop the displacement of our mass and its direction.

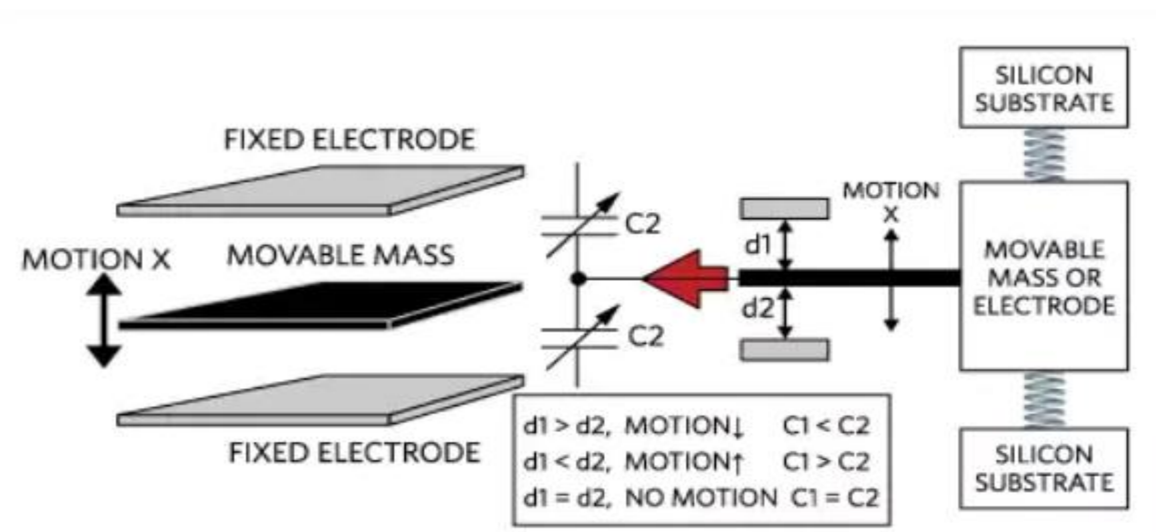

Figure 2.3 - Acceleration associated with a single moving mass.

$$C = (\varepsilon_0 \times \varepsilon_r \times A)/D \text{ (Farad)} \qquad \textbf{(Eq. 3)}$$

ε_0 = Permitted free space
ε_r = Relative material permitted between plates
A = Area of overlap between electrodes
D = Separation between the electrodes

to summarize this idea, Force causes a movement of the mass which, in return, causes a capacitance change. Now, placing multiple electrodes in parallel allows a larger capacitance, which will be more easily detected (**shown in below figure**). V1 and V2 are electrical connections to each side of the capacitors and form a voltage-divider with the center point as the voltage of our mass.

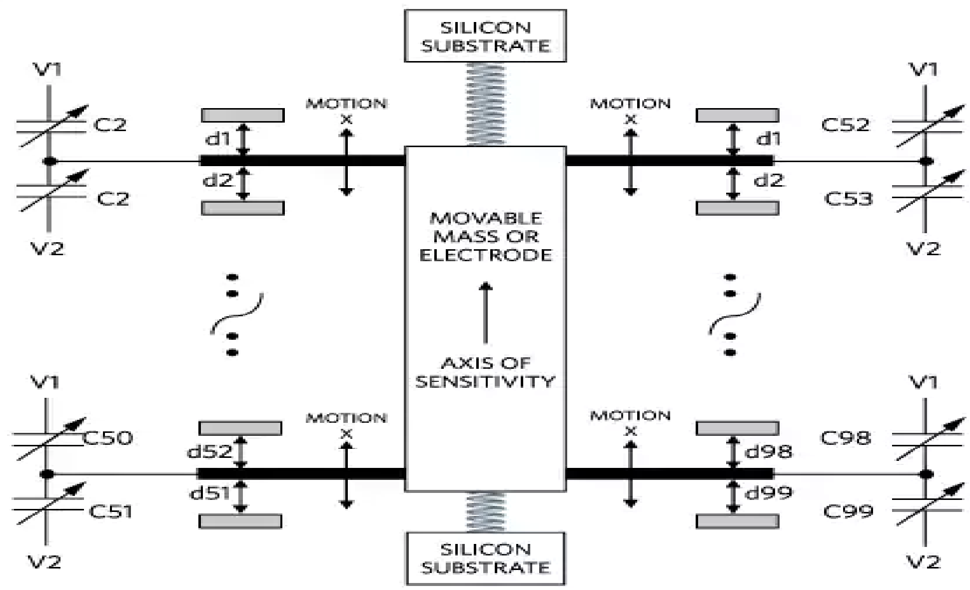

Figure 2.4 - Acceleration associated with multiple moving masses.

The analog mass voltage will go through charge amplification, signal conditioning, demodulation, and low pass filtering before it gets converted into a digital domain using a sigma-delta ADC. The serial digital bit stream from the ADC is then passed to a FIFO buffer that converts the serial signal into a parallel data stream. That parallel data stream can then be transformed using a serial protocol like I^2C or SPI before it is sent to the host for further processing (**Shown in below figure**)

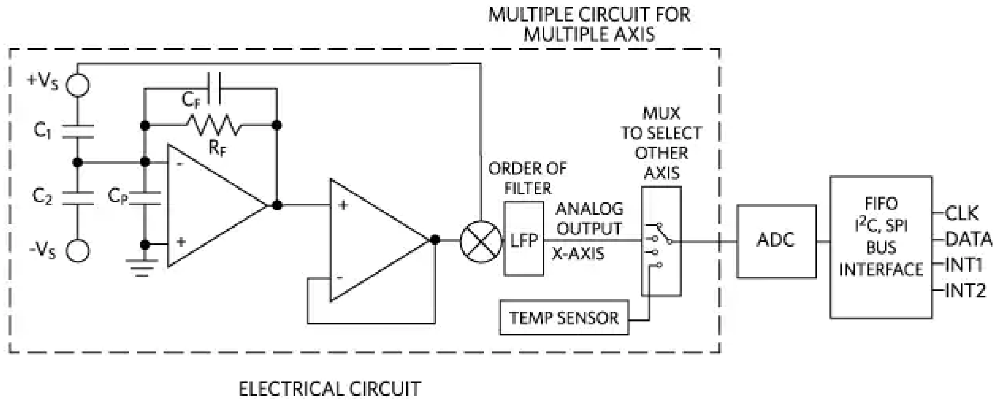

Figure 2.5 - Electrical circuit of an accelerometer.

A sigma-delta ADC is well matched for accelerometer applications because of its low signal bandwidth and high resolution. With an output value described by its number of bits, a sigma-delta ADC can be translated into "g" units for an accelerometer application very simply. The "g" is a unit of acceleration equal to the earth's gravity at sea level:

For example, if the X-axis reading of our 10-bit ADC is equal to 600 out of the available 1023 (2^{10} - 1 = 1023), and with 3.3V as the reference, we can derive the voltage for the X-axis specified in "g"with the following equation:

$$X\text{ - voltage} = (600 \times 3.3)/1023 = 1.94V \qquad \textbf{(Eq. 4)}$$

Each accelerometer has a zero-g voltage level that is the voltage that corresponds to 0g. We first calculate the voltage shifts from zero-g voltage (specified in the data sheet and assumed to be 1.65V) as:

$$1.94V - 1.65V = 0.29V \qquad \textbf{(Eq. 5)}$$

Now, to do the final conversion we divide 0.29V by the accelerometer's sensitivity (specified in the data sheet and assumed to be 0.475V/g):

$$0.29V/0.475V/g = 0.6g \qquad \textbf{(Eq. 6)}$$

By basically mounting an accelerometer in a different way (90 degrees, **shown in below figure**) we can create a 2-axis accelerometer needed for more sophisticated applications.

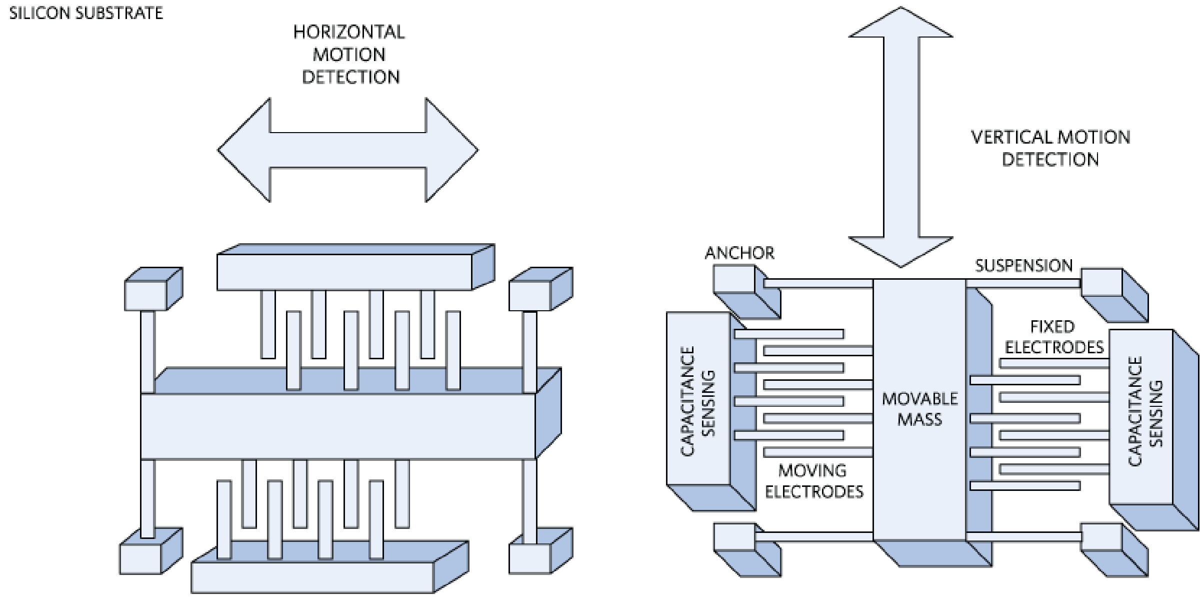

Figure 2.6 - A mechanical model of an actual accelerometer.

2.1.4 Unity game engine

Unity is a versatile cross platform game engine which works across 27 different platforms (including Windows which is where we running our simulation) developed by Unity Technologies. It is using "Direct3D" graphics API for windows & OpenGL for Linux & Macos for rendering 3D graphics & animations. It is most frequently used for improve both 2D & 3D video games for computers, mobile phones & gaming consoles. Also commonly used for computer simulations, for its sophisticated physics engine, C# scripting, drag & drop graphical rendering features.

2.1.5 Arduino platform

Arduino is increasingly becoming a popular microcontroller device for almost any electronic project unless any complex and powerful algorithms & calculations required. It is an open source computer- hardware platform consist of AT mega series processor chip as its main processing unit, GPIO pins for outside hardware interface, USB to serial interface for programming & debugging through its own easy to use programming interface called Arduino Language.

There are several versions in Arduino MCU family. Once we used to be more common among all called Arduino Uno, which will be discussed later in the later part of this project.

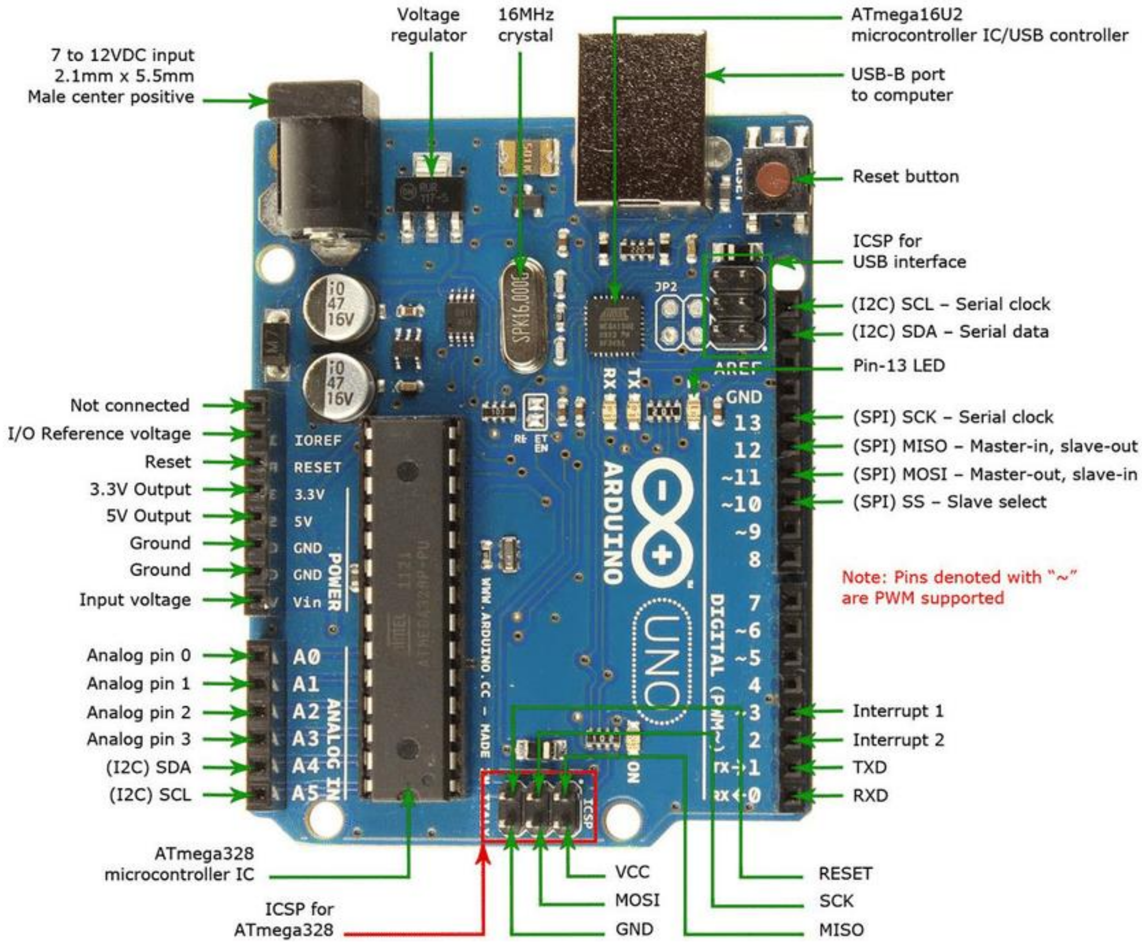

Figure 2.7– Arduino Uno Pin diagram

2.2 Related Projects

This section outlines some of the prior works & projects that are closely related with the concepts explored through the development of this VR Simulation, either in terms of the use of similar technology, or in the application of related design methodologies.

2.2.1. VOID

The VOID is a next generation immersive mixed reality entertainment franchise which currently running in three different locations across the United States centralized in its Utah VR Centre. VOID is said to be first in its kind of large-scale immersive VR project, ever initiated up until now.

Figure 2.8 – VOID simulation

VOID uses the combination of technologies such as virtual reality hardware with motion tracking, haptic feedback and special effects systems to explore and interact with virtual settings within the confines of specially designed indoor location. The Void's diverse reality experiences utilize aspects of virtual reality, haptic feedback, and other physical effects; users wear a helmet with a head-mounted display, noise cancelling headphones and a hand tracking sensor, and a haptic suit containing 22 vibrators and a computer to power the headset. The system is planned to allow its operators to freely walk through and explore a virtual world; in reality, the user is confined to a "stage" equipped with ceiling-mounted motion tracking cameras to read the user's movements. To provide physical feedback, the "stage" contains foam walls, special effects equipment such as fans, mist machines, and heat lamps, as well as props representing items such as guns and torches; all of these physical elements correspond with elements within the virtual world seen through the headset, increasing the illusion of immersion.

The original prototype of the system, which simulated the interior of a spaceship, utilized an Oculus Rift developer kit, a 10-foot long wall, and electromagnetic sensors for position tracking. Early iterations of The Void's vest system utilized an Oculus Rift Development Kit 2, Beats by Dr. Dre headphones, and a Leap Motion unit for hand tracking. These components are largely being replaced by custom-designed hardware components and systems to improve the quality of the virtual experience, and in order to overcome limitations that the original components introduced.

Figure 2.9 – unparalleled visual and body kit

One of these new components was a custom-designed head-mounted display codenamed "Rapture", which is meant to provide a higher-quality virtual reality experience in comparison to the consumer-grade hardware previously used. The Rapture headset features curved, per-eye 2K resolution OLED displays with a claimed 180-degree field of vision, a custom optics system, Bang and Olufsen headphones, and microphones for communication between players. Bretschneider personally invested $250,000 into the Rapture project. The Void's developers also plan to replace the optical tracking system with a radio-based system. The current system utilizes markers on the helmet and props that are similar to those used in a motion capture rig, which requires that the motion tracking cameras have an unobstructed view of the stage and restricts the height of the walls.

Figure 2.10 – Rapture Gloves

Figure 2.11 – VR headset

Multiple themed experiences can be intended for use on the same stage layout; experiences used at The Void's test location have included one where players explore an ancient temple, and a first-person shooter-styled mission known as "Research Facility". Outside organizations have also shown interest in having education-oriented experiences developed for The Void, including police training and nature exploration. VR experiences are being developed by The Void and by other partners, such as the University of Utah and video game studios.

2.2.2 Google Tango

Google Tango (also known as Project Tango) is an advanced computer vision augmented reality platform, initiated by Advanced Technology and Projects (ATAP), a skunkworks division of Google. The platform itself uses various technologies such as computer vision, motion tracking, depth perception & some image processing techniques to enable mobile devices to detect their position relative to the world around them without using GPS or any other external signals. This platform also gives opportunity to the developers to create applications with user experiences of indoor navigation, 3D mapping, physical space measurement, environmental recognition, augmented reality, and windows into a virtual world.

Figure 2.12 – Google tango

Tango is unique from other emerging 3D-sensing computer vision products, in that it's designed to run on a standalone mobile phone or tablet and is chiefly concerned with deciding the device's position and orientation within the environment.

The software works by integrating three types of functionality:

I. Motion-tracking: using visual features of the environment, in combination with accelerometer and gyroscope data, to closely track the device's movements in space

II. Area learning: storing environment data in a map that can be re-used later, shared with other Tango devices, and enhanced with metadata such as notes, instructions, or points of interest

III. Depth perception: detecting distances, sizes, and surfaces in the environment

Together, these generate data about the device in "six degrees of freedom" (3 axes of orientation plus 3 axes of motion) and detailed three-dimensional information about the environment.

Project tango was also the first project to graduate from Google X in 2012.

Applications on mobile devices use Tango's C and Java APIs to access this data in real time. In addition, an API is also provided for integrating Tango with the Unity game engine; this enables the rapid conversion or creation of games that allow the user to interact and navigate in the game space by moving and rotating a Tango device in real space. These APIs are documented on the Google developer website.

CHAPTER 3 - CONCEPTUAL FRAMEWORK

3.1 Virtual Reality

We know the world through our senses and perception systems. In school we all learned that we have five senses: taste, touch, smell, sight and hearing. These are however only our most obvious sense organs. The truth is that humans have many more senses than this, such as a sense of balance for example. These other sensory inputs, plus some special processing of sensory information by our brains ensures that we have a rich flow of information from the environment to our minds. Everything that we know about our reality comes by way of our senses. In other words, our entire experience of reality is simply a combination of sensory information and our brains sense-making mechanisms for that information. It stands to reason then, that if you can present your senses with made-up information; your perception of reality would also change in response to it. You would be presented with a version of reality that isn't really there, but from your perspective it would be perceived as real. Something we would refer to as a virtual reality. Virtual reality is the term used to describe a three-dimensional, computer generated environment which can be explored and interacted with by a person. There are a wide variety of applications for virtual reality which include:

- Architecture
- Sport
- Medicine
- The Arts
- Entertainment

Virtual reality can lead to new and exciting discoveries in these areas which impact upon our day to day lives.

Virtual reality simulation is the use of 3D objects and environments to create immersive and engaging learning experiences. The principle of virtual reality e-learning is to impart, practice and check a user's knowledge using interactive scenarios and environments to reflect real-life situations.

Using 3D technology, which we refer to as Real World Environments, we can create any environment or situation. Through this immersive technology, the world of learning is moving to an entirely new level. In addition to the standard e-learning methods of viewing and hearing information, virtual reality simulation provides a truly interactive experience. Users can move freely around the environment, interact with objects, carry out tests, and make decisions and mistakes until they have mastered the subject. By letting learners practice in a virtual environment you will not only be able to see what they've learnt but also they're approach and thought process to a problem.

Positional tracking is very important towards achieving immersion and presence in virtual reality. There are many ways to achieve positional tracking, with pros and cons for each solution. There are many ways of positional tracking technologies which could be used for virtual reality. There are many factors to weigh:

- Tracking accuracy required
- Refresh rate required
- Whether the objects to be tracked are well known or can change from time to time
- Whether the objects to be tracked are rigid or flexible
- The desired tracking area
- Whether tracking needs to be indoor or also outdoor, where lighting conditions might be more challenging
- Cost, power consumption and computational power available for tracking

The tracking devices are the main components for the VR systems. They interact with the system's processing unit. This relays to the system the orientation of the user's point of view. In systems which let a user to roam around within a physical space, the locality of the person can be detected with the help of trackers, along with his direction and speed.

The various types of systems used for tracking utilized in VR systems. These are as follows: -

- A six degree of freedom can be detected (6-DOF)
- Orientation consists of a yaw of an object, roll and pitch.
- These are nothing but the position of the objects within the x-y-z coordinates of a space, however, it is also the orientation of the object.

These however emphasizes that when a user wears a HMD then as the user looks up and down; left and right then the view also shifts. Whenever the user's head tilts, the angle of gaze changes. The trackers on the HMD describe to the CPU where you are staring while the right images are sent back to the screen of HMD.

All tracking system consists of a device that is capable of generating a signal and the signal is detected by the sensor. It also controls the unit, which is involved in the process of the signal and sends information to the CPU. Some systems ask you to add the component of the sensor to the user (or the equipment of the user's). If this takes place, then you have to put the signal emitters at certain levels in the nearby environment. Differences can be easily noticed in some systems; with the emitters being worn by the users and covered by sensors, which are attached to the environment. The signals emitted from emitters to different sensors can take various shapes, including electromagnetic signals, optical signals, mechanical signals and acoustic signal.

Figure 3 – Virtual Reality Game

3.2 Natural interaction techniques

We consider natural interaction techniques to be those techniques that mimic real-world interaction by using body movements and actions that are similar to those used for the same task in the physical world. Clearly, there are varying degrees of fidelity for natural interaction.

These natural interaction techniques make games easier to play by leveraging re world experiences, similar to embodied interaction. On the other hand, there are also arguments for the use of non-natural interaction techniques that allow users to overcome limitations of their cognitive, perceptual, and motor capabilities.

For example, in the virtual reality (VR) domain, many non-natural interaction techniques are used to compensate for limited human motor capabilities by allowing users to access distant objects or travel in ways not naturally possible. Furthermore, some natural interaction techniques have been demonstrated to lack precision compared to non-natural techniques due to physical inadequacies of the human body, such as 3D pointing techniques suffering from hand jitter.

Player performance influences player enjoyment, which is the single most important goal for a video game, since players will not continue to play a game they do not enjoy. Player performance is additionally important because players will experience anxiety if the challenge of a game proves much greater than their own ability to perform.

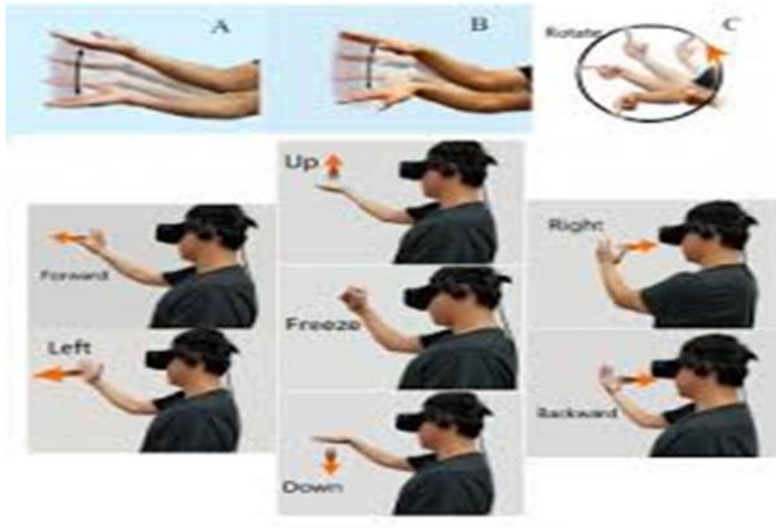

Figure 3.1

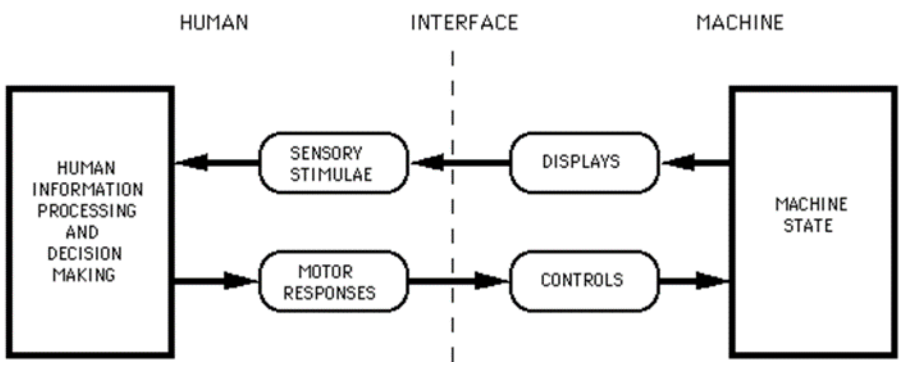

Figure 3.2

Most of the obstacles making the long awaited switch from a 2D computing environment to a 3D are diminishing. Fast workstations with high-end graphics cards are affordable, animation techniques are getting more complex and realistic, new input devices are emerging.

CHAPTER 4 - PROJECT DEVELOPMENT

In this section, details are documented about how the project is developed, including the technology that is incorporated into the system, how the prototypes are tested, and the targets for development.

4.1. Feasibility Study & Analysis

Before we started the project development, we have gone through a cycle of feasibility study & analysis over various technologies to fulfill our project requirement.

As we earlier discussed in section 1.4 in this document we initially divided this project into three phases in order to achieve our objective. The hardest challenge we face in this project was to achieve the first phase from those three. That is to capture the real time position & the orientation of the player with sub centimeter accuracy.

We have analyzed and tried various techniques to complete this phase, including trilateration and various death reckoning techniques. Let's summarize the analyze we did on each of those techniques.

4.1.1 Trilateration using ultrasonic transducers

When it comes to indoor localization, trilateration is the most common technique in industry practices. The overall technology background was earlier discussed in section 2.1.2 in this document. Now let's focused about trilateration using ultrasonic transducers in particular, that we've tried for complete our first phase of the project.

Ultrasonic transducers are good option for using as a medium for distance measuring in trilateration algorithm, since it has very low propagation delay, that will reduce the syncing errors & easy installation process compared with other mediums.

The transmitter sends out an ultrasonic sound wave and simultaneously a signal is sent via cable which tells the receiver to start count the time. When the sound wave reaches the receiver, it stops counting. By knowing the speed of sound this distance can be calculated with the relation distance=speed*time.

In this technique transducer, installation can be made in two possible ways. One way is to have a single transmitter that act as some mobile node & multiple receivers in fixed location. Second way is the other way around, have a single receiver and multiple fixed-location transmitters. Since we want to located only one mobile node, the first way is the preferred one for us & move along with it.

In order to achieve good result, we had to consider among different ultrasonic transmitter parameters before buying one for the project, such as It's signal frequency, strength & Attenuation parameters etc.

In order to get a reliable real time location tracking, we have to choose a transmitter with high frequency but also low enough to neglect the overlapping effects of the signals caused if the mobile nodes moving fast towards the transmitter. 40 Khz seems to be the best tradeoff for us considering these facts. When it comes to attenuation it is the price factor influenced a lot. The less attenuation characteristics we opt for the more price we have to pay.

However, due to following reasons we have to deprecate the trilateration techniques

1. Difficulty & the lack of knowledge to process & modulate ultrasonic signals transmitted by ultrasonic transmitters

2. In order to calculate distance by measuring TOF measurement, all transmitters should be perfectly synchronized & the real time data received from the receiver should be coordinated with all the transmitters. Other ways there will be errors

3. Since ultrasonic signals are not having wide omnidirectional range, it is not always sure that the receiver will always receive the signal send by the transmitter.

4.1.2. Death reckoning using ST Sensor tile kit

Sensortile is a SOC (system on chip) with powerful arm cortaxm4 processor which capable of running complex instructions. Which capable of running complex instructions. Which also integrates on chip high precise memes sensors for IMU measuring and magnometer and humidity, temperature sensors.

Due to complex ICSP programming interface and the lack of field expertise, we were unable to program the chip on order to retrieve its sensor data.

Figure 3.3 – Sensortile kit and expansion board

4.2 Initial System design

Once we fixed our location tracking phase, we started building our initial prototype by using MPU 6050 circuit board for sensing inertial measurements & Arduino Uno Revision3 MCU unit for communication interface In between Inertial sensor module(MPU6050) & the Unity platform. Below figure shows the overall high-level architecture of our system design.

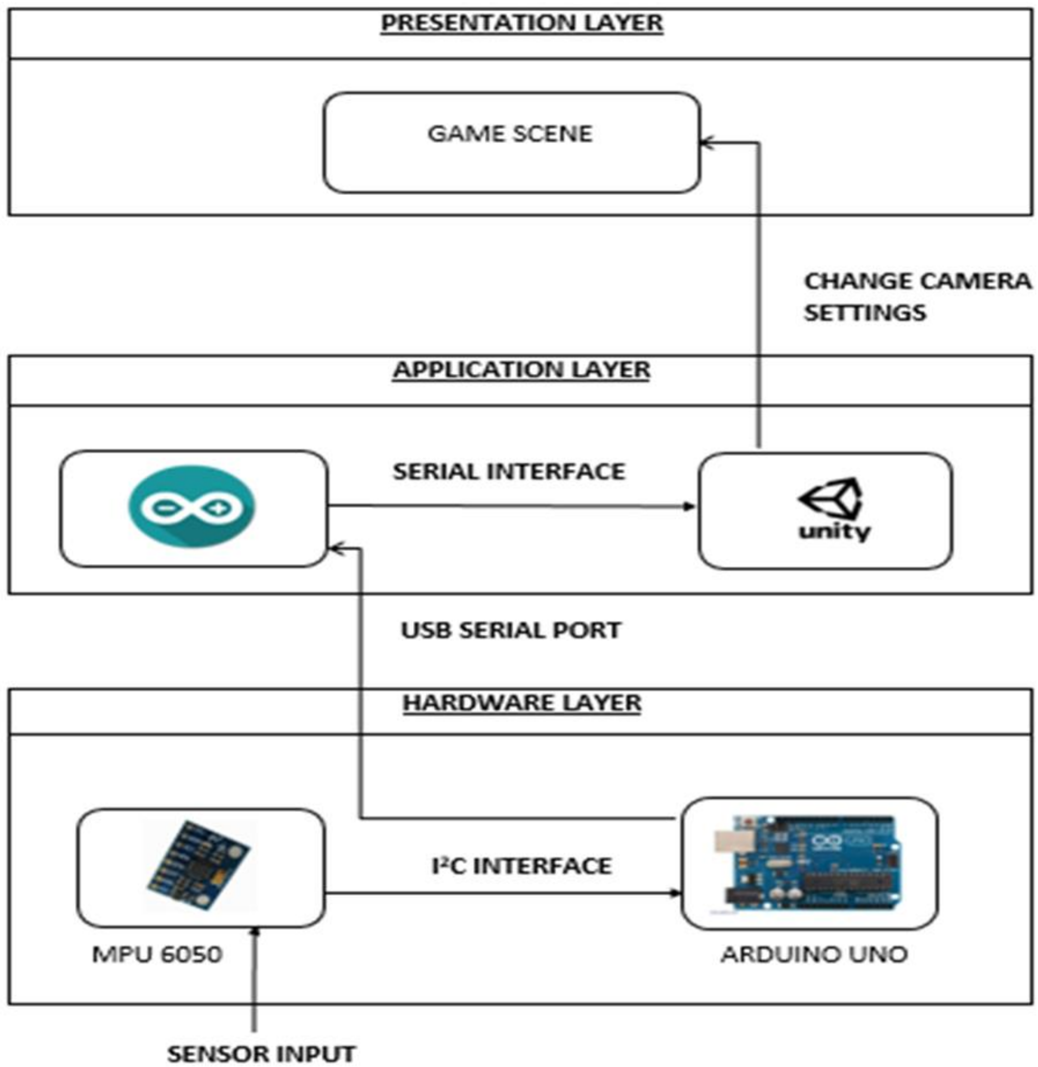

Figure 3.4 – System architecture

4.2.1. Hardware installation

Since the MPU 6050 module uses I2C (Inter-integrated Circuit) protocol for data transmission, we have to use Arduino's I2C communication protocols in order to receive data from MPU module.

Since Arduino Uno's Shared Clock & Shared data line for the I2C protocols multiplexed with its Analog pin 5 & Analog pin 4, we have to use those GPIO pin for establish the a successful communication.

MPU 6050 Pins	Arduino Uno R3 pins
Vcc	To 5V pin of the Arduino
Gnd	To one of the Arduino's Gnd pin
SCL	Analog pin 5 (A5) of Arduino
SDL	Analog pin 4 (A4) of Arduino
INT	Pin 2 (not necessary for our purpose)

Table 1

4.2.1.1 MPU 6050

MPU 6050 is the core-sensing device of our project. It contains a 3 axis accelerometer, a temperature sensor & a 3-axis gyroscope. This circuit module sends data using I2C protocol and works with 5V logic reference. We use Arduino Uno board for reterive real time data from this module since it works well with Arduino platform. The source code and the data communication mechanism will be discussed in the later part of this document.

MPU 6050: reading the raw values

As "Jeff Rowberg" said in his own i2cdevlib forums

"The accelerometer and gyroscope measurements are explained in the MPU-6050 datasheet in the GYRO_CONFIG and ACCEL_CONFIG register descriptions (sections 4.4 and 4.5 on pages 14 and 15). The scale of each depends on the sensitivity settings chosen, which can be one of +/- 2, 4, 8, or 16g for the accelerometer and one of +/- 250, 500, 1000, or 2000 deg/sec for the gyroscope. The accelerometer produces data in units of acceleration (distance over time2), and the gyroscope produces data in units of rotational velocity (rotation distance over time). The output scale for any setting is [-32768, +32767] for each of the six axes. The default setting in the I2Cdevlib class is +/- 2g for the accel and +/- 250 deg/sec for the gyro. If the device is perfectly level and not moving, then:

X/Y accel axes should read 0

Z accel axis should read 1g, which is +16384 at a sensitivity of 2g

X/Y/Z gyro axes should read 0

noise/error, and the gyros will also not read exactly 0 for the same reason (noise/error)."

So to convert raw values into meaningful acceleration & angular velocity values, first we have to identify the sensitivity of our sensor. Since we didn't change the default settings, it was remained +/-2G for accelerometer & +/- 250deg/s for gyroscope. Since the output range of the accelerometer is -32768 to +32767 & the sensitivity is +/- 2g, we can calculate and say 1G we will read 16384 and similarly for gyro 1degree/sec we will read 131.07 (can neglect the decimal points).

4.2.1.2 Arduino Uno 8 bit MCU

In our design Project, we used Arduino just to use as a communication interface between MPU 6050 Module & Unity platform. It's analog pin 4 and 5 GPIO pins used for I2C's clock and Shared data bus line. It receiver's data through this shared line & by it's serial USB code it send all those data serially to our computer, which we used for unity simulation.

4.2.2 Software Design

In our VR stimulation, software design plays the major role. The only electronic interface it contains is to measure inertial measurement from the outside world by using IMU sensor interface. Apart from that all the remaining work, from Data processing to scripting to virtual simulating, all the remaining part was highly relying on software design.

4.2.2.1 Arduino Program

The Arduino sketch (source code) was developed for fetching accelerometer & gyroscope values in and send it over to unity via its serial interface in real time.

```
/* code begins from
below…………………………………………………………………………………………………
. */


#include<Wire.h>  //Arduino library that contains I2C Communication methods

const int MPU6050 = 0x68;  //Device address, by default MPU 6050 assigned to 0x68

int16_t AcX, AcY, AcZ, Tmp, GyX, GyY, GyZ, velX, velY, velZ;

/*16-bit integer variables for hold acceleration, gyro values along three axis & to hold
temperature data */

int data;


void setup() {

  Wire.begin(); // starting I2C Communication
```

```
  Wire.beginTransmission(MPU6050); //initializing communication with our MPU module
  Wire.write(0x6B);
  Wire.write(0); //flag the MPU to start sending its data
  Wire.endTransmission(true); //stop sending data
  Serial.begin(115200); //initialize computers serial interface with 115 200 bytes/sec baud rate
}
void loop() {
  GetMPU_Value();  //function for fetch MPU sensor values
  SetUniity(); //send those MPU values to unity via computers serial interface
}
void GetMPU_Value(){
  Wire.beginTransmission(MPU6050);
  Wire.write(0x3B); //accel address //start writing from accelX from all other data
  Wire.endTransmission(false); // asking to continue write data
  Wire.requestFrom(MPU6050, 14, true); //up until 14 bytes of data received

/* splitting the MPU's packetized data into each individual components */
  AcX = Wire.read()<<8| Wire.read();
  AcY = Wire.read()<<8| Wire.read();
  AcZ = Wire.read()<<8| Wire.read();
  Tmp = Wire.read()<<8| Wire.read();
  GyX = Wire.read()<<8| Wire.read();
  GyY = Wire.read()<<8| Wire.read();
  GyZ = Wire.read()<<8| Wire.read();

/* updating the velocity by repeatedly adding up acceleration value with a small sampling time
of 10 milliseconds */
```

```
 velX += (AcX/16384)*9.8*0.01;
  delay(10);
  velY += (AcY/16384)*9.8*0.01;
  delay(10);
  velZ += ((AcZ-16384)/16384)*9.8*0.01;
  delay(10);
}
/* print those acceleration & angular velocity values to serial interface, so that our unity script
will later catch up those values in real time */
void SetUniity(){
if (Serial.available()){

  data = Serial.read();

    if(data == 'a') Serial.println(velX);
    if(data == 'b') Serial.println(velY);
    if(data == 'c') Serial.println(velZ);
    if(data == 'd') Serial.println(GyX);
    if(data == 'e') Serial.println(GyY);
    if(data == 'f') Serial.println(GyZ);
  }
}
/* end of the program……………………………………………….  */
```

4.2.2.2 Unity Scene

This unity is what we're going to simulate. Since its highly functional game engine provide all existing graphical API's, whatever the scene we've designed in its graphical interface will be ready to simulate based on our script. Below figures captures screenshots of our simulation,

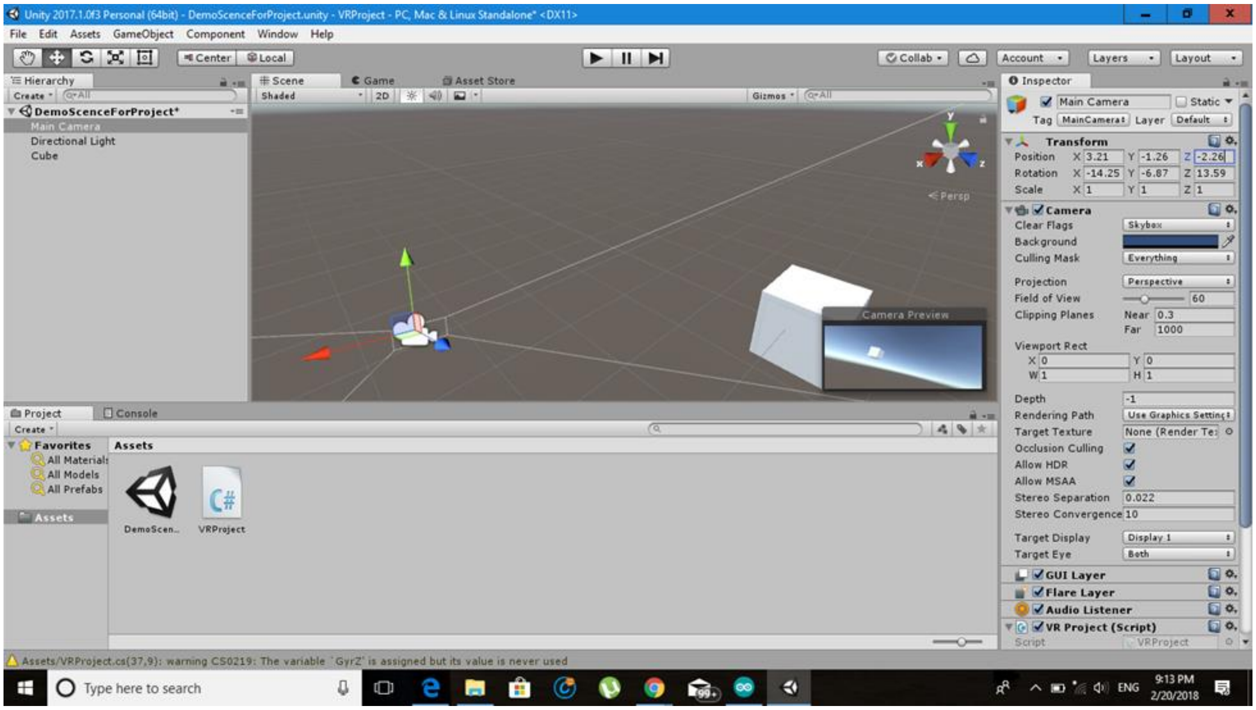

Figure 4.1 – unity interface

In our unity scene, we created a C# Script under the Assets menu in above picture. This C# Script contains script that will basically tell our camera object to move based on our MPU Sensor readings, which in turns will create a nice Virtual Simulation within its scene environment.

In order to work our camera object, we have to drag our C3 script that was named “VR Project” in above picture drop it to main camera list menu in the rift corner menu bar in our unity program.

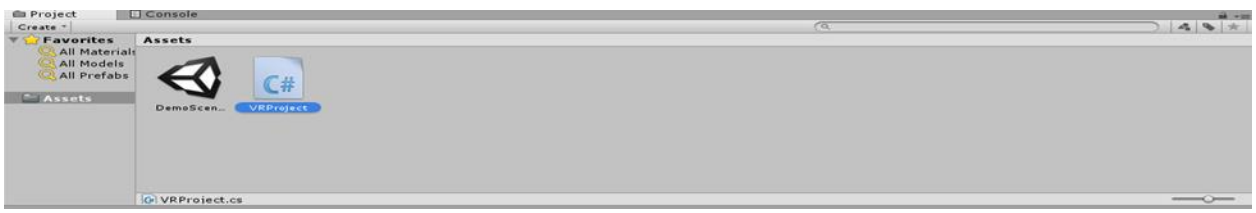

Figure 4.2 – C# script we created to control the view of our simulation

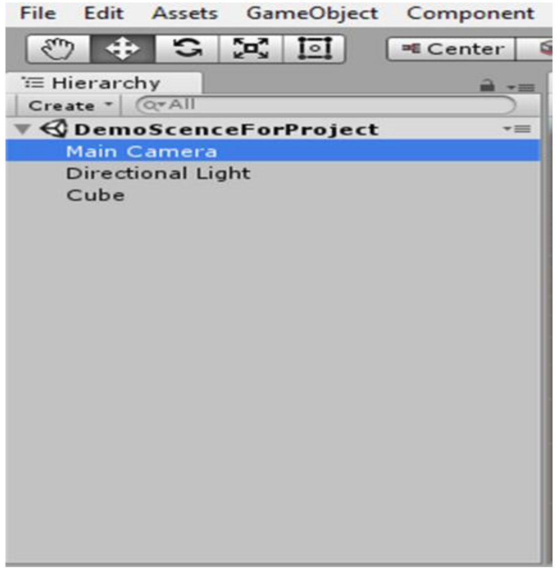

Figure 4.3 – main camera objects where we attached our C# Script by drag & drop it.

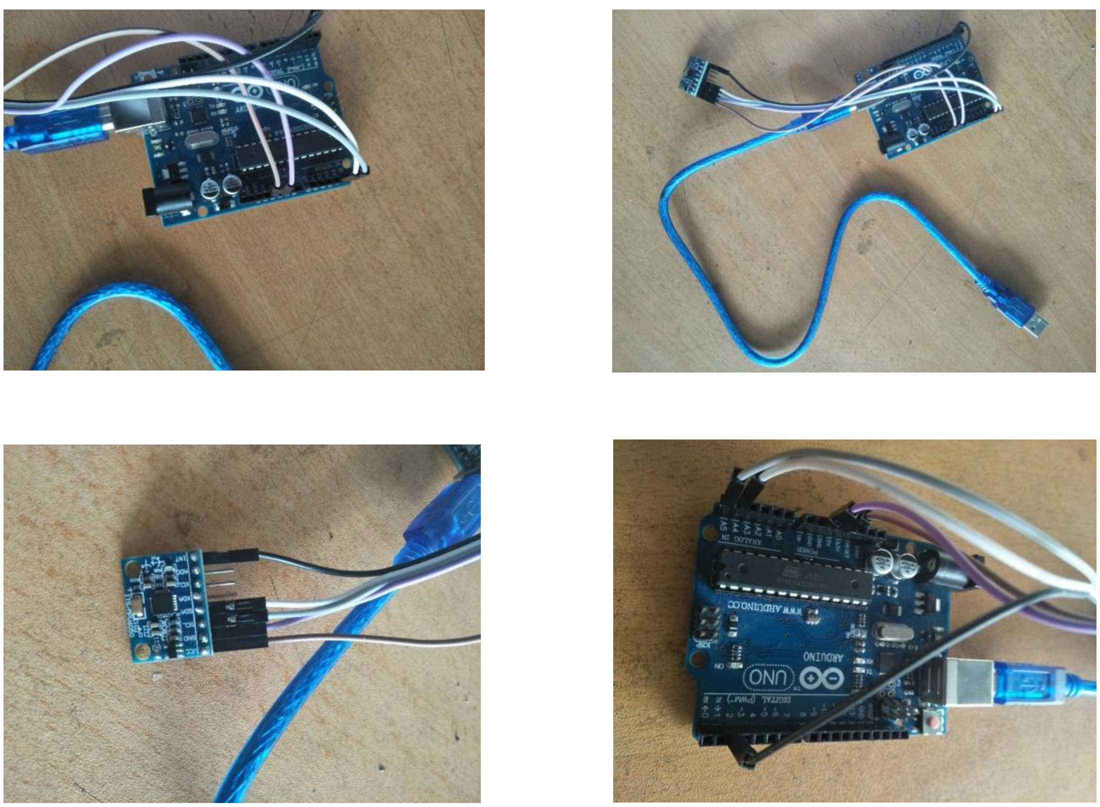

figure 4.4 – final outcome of the project

4.2.2.3 C# Script for Unity scene

This C# Script are the instructions to our VR simulation which tells our unity programs to change the camera view of the simulation based on the MPU sensor value that we processed.

```
/*  Start of the code ……………………………………..        */

using System.Collections;

using UnityEngine;

using System.IO.Ports;


public class VRProject : MonoBehaviour {

        public float y;

        public float z;

        public float x;

        public float posX;

        public float posY;

        public float posZ;

        SerialPort serial = new SerialPort("COM3", 115200);


        void Start () {

                serial.Open ();

                serial.ReadTimeout = 2000;

}


        void Update () {

                if (!serial.IsOpen)

                        serial.Open ();


                serial.Write ("a");
```

```
                    float velX = int.Parse (serial.ReadLine ());
                    serial.Write ("b");
                    float velY = int.Parse (serial.ReadLine ());
                    serial.Write ("c");
                    float velZ = int.Parse (serial.ReadLine ());
                    serial.Write ("d");
                    float GyrX = int.Parse (serial.ReadLine ());
                    serial.Write ("e");
                    float GyrY = int.Parse (serial.ReadLine ());
                    serial.Write ("f");
                    float GyrZ = int.Parse (serial.ReadLine ());


                    x += GyrX / 2620;
                    y += GyrY / 2620;
                    z += GyrX / 2620;
          /*change the position consecutively by keep adding up (velocity * sampling time(1/200
Sec))  */
                    posX += velX / 200;
                    posY += velY / 200;
                    posZ += velZ / 200;


transform.localEulerAngles = new Vector3 (-x, -y, -z);
transform.localPosition = new Vector3(posX, posY, posZ);
          }
}


/*end of the code……………*/
```

4.2.3 Network Configurations

I2C Communication

This communication method defines the way, which microcontroller and the modules communicate with each other.

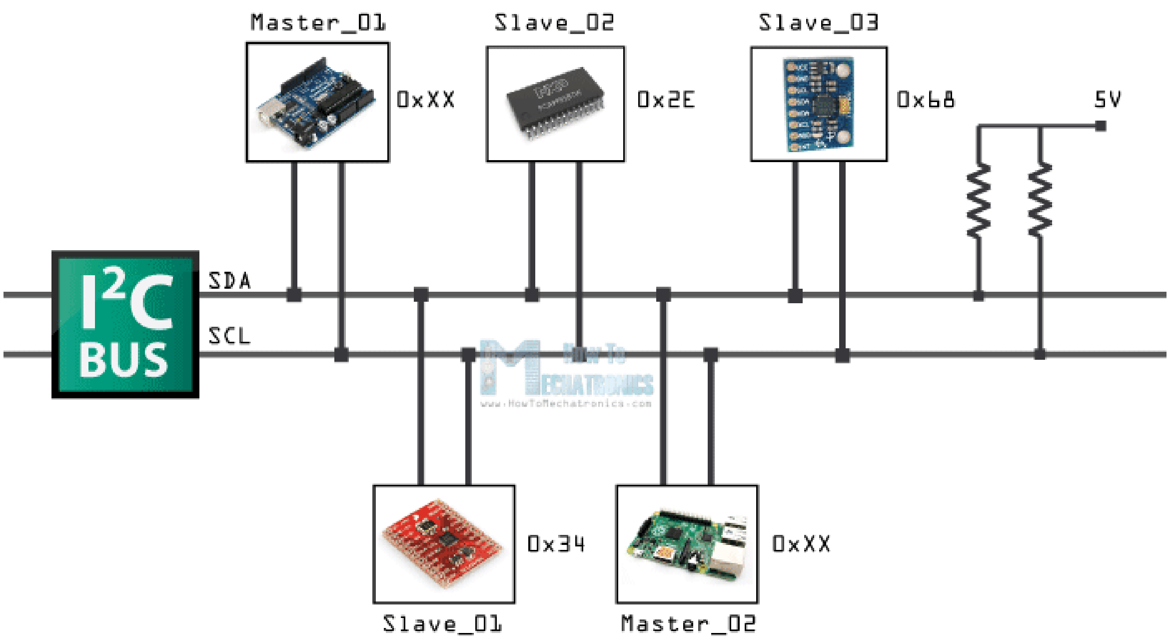

Figure 4.6 – I2C communication

An active low voltage triggers the communication and this consists of two channels. One is the shared clock line, which synchronizes the devices, and the other is the shared data line, which drives the data.

Arduino chip works as a master and the module works as a slave. This always refers to the address of each and every component.

Serial Communication

Serial Communication method emphasizes the way, which Arduino processor and the machine communicates with each other. This has only one data channel, which transfers data as a bit stream one after the other.

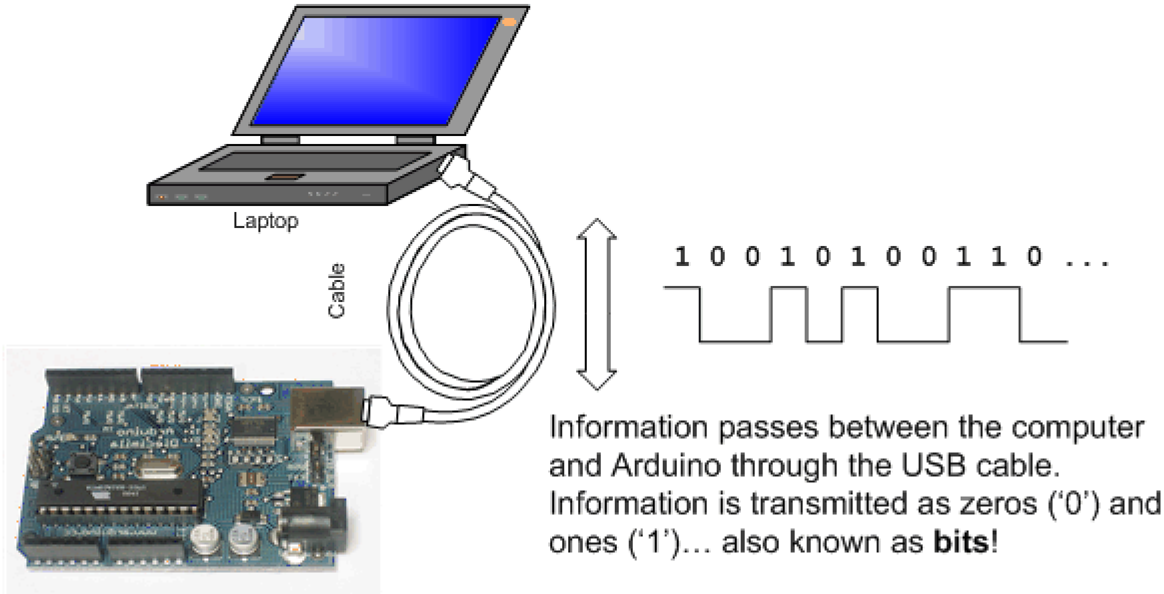

Figure 4.7 – Serial communication

All these Arduino elements are connected through Bluetooth modules.

4.3 Testing and troubleshooting

Several test cases were tested during the testing phase. Data sheets were referred to find the relevant values such as bit rates to be transferred, baud rates and the sensitivity values. To select the sampling rate we choose a range from 10 to 100 ms. Then we have inputted all the values within the range to achieve the most stable point. The range should be corrected. It should be replaced by the range of o to 100. Finally add this part, ultimately we could observe that at the sampling rate of 5 the most stable point could be obtained.

CHAPTER 5 - POST DEVELOPMENT

5.1 Future Considerations

Virtual reality technology has seen rapid developments in recent years and this is most apparent in the architectural, engineering and construction industry. Every design will soon be made using virtual reality; enabling the user to fully immerse himself in a 1:1-scale, 3D (BIM) model which can be manipulated and provides an incredibly accurate sense of presence in a space that's yet to be built.

- Military populations
- Space exploration
- Geriatric populations
- Pain due to cancer and other chronic injuries

5.2 Conclusion

The whole project was driven on a prototype based working model which visualized a VR simulation. It was developed up to some extent within our capabilities. The whole intergraded system regarding to the VR concept can be enhanced to a military training simulation in the future.

CHAPTER 6 – REFERENCES

1) Carlos Medina *, Jose´ Carlos Segura and A´ ngel De la Torre, 2013. Ultrasound Indoor Positioning System Based on a Low-Power. sensors, p. 26.

2) Muhammad Farooq-i- Azam, M. N. A., n.d. Location and Position Estimation in Wireless Sensor Networks. Pakistan: s.n.

3) Nasrullah Pirzadaa*, M Yunus Nayanb , Fazli Subhanc M Fadzil Hassand, Muhammad, 2013. Device-free Localization Technique for Indoor Detection. Malaysia, Procedia - Social and Behavioral Sciences 129 (2014) 422 – 429.

4) Seco, A. J. a. F., 2005. Ultrasonic Localization Methods for Accurate Positioning,Madrid (Spain): s.n.

5) Johnson, Leif (13 March 2016). "Forget 360 Videos, Photogrammetric Virtual Reality Is Where It's At - Motherboard". Motherboard. Retrieved 20 March 2017.

6) Rosenberg, Louis (1992). "The Use of Virtual Fixtures As Perceptual Overlays to Enhance Operator Performance in Remote Environments.". Technical Report AL-TR-0089, USAF Armstrong Laboratory, Wright-Patterson AFB OH, 1992.

APPENDIX & GLOSSARY

VR - Virtual reality

HMI - Human Machine Interaction

IMU - Inertial Measurement Unit

IPS - Indoor Positioning System

GPS - Global Positioning System

TOF -Time Of Flight

Unity - Versatile cross platform game engine

Arduino board