

## Abstract

This research will offer another way for developers to place the user within Virtual Reality, showing and explaining key points which can make the user feel more immersed into the environment using full body tracking.

## Acknowledgements

I am grateful to all of those with whom I have had the pleasure of working with throughout my time within Peterborough Regional College and University Centre Peterborough, the friends I have made I will carry with me beyond my time here and I would like to say thank you to all of those I have had the pleasure of meeting.

There are a few individuals I would like to thank for their contribution to my work this final year. Firstly, I’d like to thank Mr Don MacSween, who’s always made time for me and always offered support when needed throughout my time at college and University.

Finally, I would like to say a huge thank you to Mr Jamie Myland, words alone cannot describe the gratitude I have for the time, help and guidance he has given me over these last 5 years.

**Table of Contents**

[Abstract 2](#_Toc9248011)

[Acknowledgements 2](#_Toc9248012)

[Chapter 1: Introduction 1](#_Toc9248013)

[Background 1](#_Toc9248014)

[Justification 1](#_Toc9248015)

[Aims 2](#_Toc9248016)

[Objectives 2](#_Toc9248017)

[Problem 4](#_Toc9248018)

[Chapter 2: Research 5](#_Toc9248019)

[Literature Review 5](#_Toc9248020)

[Current Virtual Reality Systems Available 7](#_Toc9248021)

[Current Full Body Tracking Systems 10](#_Toc9248022)

[Justification of device for full body tracking 16](#_Toc9248023)

[Current available platforms for 3D Environment development 17](#_Toc9248024)

[Justification of platform used for development 26](#_Toc9248025)

[Chapter 3: Methodology 27](#_Toc9248026)

[Overview 27](#_Toc9248027)

[Proof of Methodology 28](#_Toc9248028)

[Chapter 4: Implementation 29](#_Toc9248029)

[Lower Body Tracking 29](#_Toc9248030)

[Tracking the hands and head 32](#_Toc9248031)

[Goalie Equipment 35](#_Toc9248032)

[Chapter 5: Findings & Discussion 43](#_Toc9248033)

[Findings 43](#_Toc9248034)

[Discussions 46](#_Toc9248035)

[Chapter 6: Analysis 49](#_Toc9248036)

[Chapter 7: Conclusion & Future Work 51](#_Toc9248037)

[Conclusion 51](#_Toc9248038)

[Future Work 52](#_Toc9248039)

[Appendix 53](#_Toc9248040)

[Section 1 Part A: List of Supervisor Minutes 53](#_Toc9248041)

[Section 1 Part B: Gantt Chart (Before and After) 55](#_Toc9248042)

[Section 2: Articles 56](#_Toc9248043)

[References 58](#_Toc9248044)

[Figure 1: Facebook Full Body Tracking 5](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248086)

[Figure 2: HTC Vive System 7](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248087)

[Figure 3: HTC Vive Pro System 8](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248088)

[Figure 4: Oculus Rift System 8](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248089)

[Figure 5: PlayStation VR System 9](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248090)

[Figure 6: Samsung Odyssey Windows Mixed Reality System 9](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248091)

[Figure 7: Xbox 360 Kinect 10](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248092)

[Figure 8: HTC Vive Tracker 11](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248093)

[Figure 9: OptiTrack System Layout Diagram 12](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248094)

[Figure 10: Perception Neuron Full Body Tracking System 13](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248095)

[Figure 11: PrioVR System 14](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248096)

[Figure 12: TVico VR Sensor 15](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248097)

[Figure 13: Unity 3D 18](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248098)

[Figure 14: Unity Development Statistics 19](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248099)

[Figure 15: Unreal Engine 22](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248100)

[Figure 16: Xbox 360 Kinect V1 29](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248101)

[Figure 17: Diagram of the Kinect V1 29](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248102)

[Figure 18: Driver4VR Mapping Skeleton to User 31](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248103)

[Figure 19: 5 Sensors (head, hands, hips and feet) 31](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248104)

[Figure 20: HTC Vive Headset 32](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248105)

[Figure 21: HTC Vive Controllers 33](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248106)

[Figure 22: HTC Vive Controller without casing 33](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248107)

[Figure 23: HTC Vive Base Stations 34](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248108)

[Figure 24: HTC Vive Lighthouse without casing 34](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248109)

[Figure 25: Goalie Equipment - Catcher 35](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248110)

[Figure 26: Mesh Colliders 36](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248111)

[Figure 27: Construction of the goal 37](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248112)

[Figure 28: Sample code of goal detecting script 37](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248113)

[Figure 29: Shooting area 38](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248114)

[Figure 30: Sample code for gathering target 39](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248115)

[Figure 31: Generating Puck Velocity 40](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248116)

[Figure 32: Firing the puck 40](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248117)

[Figure 33: Shooting point 41](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248118)

[Figure 34: The puck 41](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248119)

[Figure 35: Scoreboard 42](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248120)

[Figure 36: End of game message 42](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248121)

[Figure 37: Kinect Projects 47](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248122)

[Figure 38: Kinect V1 for Windows 48](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248123)

[Figure 39: Kinect V2 for Windows 48](https://studentpeterboroughacuk-my.sharepoint.com/personal/p110075425_student_peterborough_ac_uk/Documents/Peterborough%20-%20University/Year%203/Dissertation/Documentation/Final%20Report/Final%20Report.docx#_Toc9248124)

# Chapter 1: Introduction

## Background

The aim of this project is to develop a Virtual Reality experience with the added functionality of full body tracking; something current Virtual Reality games do not offer users officially or with easy convenience to use; but the technology is available. However, it has yet to become an industry standard for developers due to the high prices and infeasible methods of application. *(UploadVR, 2019)*

This will be achieved by using a platform that enables development of Virtual Reality environments, with the capability of allowing quick development for a lone developer, in conjunction with current motion tracking technologies.

## Justification

One of the main issues currently with Virtual Reality is the immersion level; visually the systems available to the users are effective at making the user feel as if they’re within the environment. However, once the user begins to interact with the environment that’s where the immersion level tends to bottleneck the experience; this is due to the system not being able to fully portray the user within the environment *(Kirill karev, 2018).* This research will offer another way for developers to place the user within Virtual Reality, showing and explaining key points which can make the user feel more immersed into the environment.

## Aims

* To analyse the different platforms available that will allow a lone developer to build a game in a Virtual Reality environment.
* To analyse the different methods of full body tracking in Virtual Reality and establish which one offers the most effective, cheapest and robust tracking system.

## Objectives

* Establish the best available platform that will allow for rapid Virtual Reality development
* Establish the most effective and currently available full body tracking systems for Virtual Reality
* Design 3D models for virtual reality that are effectively optimised for this type of environment
* Get all hardware to work seamlessly and increase the user’s immersion in a Virtual Reality Environment
* Develop the program from start to finish with an Object-Oriented concept as the core structure
* Test the artefact against the aims of the project

**Outcomes:**

* Develop an application with a consistent rate of 90 frames per second
* The finished application will accurately track the position of both upper and lower sections of the user’s body
* When a puck is generated, the puck shooter will select a position within the defined shooting area and fires the puck at the corresponding coordinates
* The application will offer collision detection between the puck and the user’s body, so that the user can block the path of the puck stopping it from going into the net

## Problem

Virtual Reality alone has yet to reach its full potential, currently the Virtual Reality experience means being tied up to an expensive PC and having sensor pods strapped to certain parts of the user’s body; not very effective for getting a fully immersive experience that Virtual Reality has the potential to give.

An article by Tony, an AR and VR Developer, was published on Medium; a publishing platform developed for offering a hybrid collection of journalism of amateur and professional developers. In the article he lists current problems with Virtual Reality from a user's perspective; that being:

* The player can’t see their body
* Players can’t interact in a natural way
* Motion sickness can be caused by this unnatural interaction

(Medium, 2017)

Visually the systems available to the users are effective at making the user feel as if they’re within the environment. However, once the user begins to interact with the environment that’s where the immersion level decrease; this is due to the system not being able to fully portray the user within the environment (Kirill karev, 2018).

# Chapter 2: Research

## Literature Review

An initial 2 articles can be found in the Appendix, under Section 2 “Articles”.

Virtual Reality is currently in a major transition, the technology behind the headsets has been in the market for the majority of 5 years; companies have established guidelines and a better understanding of the technology and have reached a point where the next generation needs to be revolutionised.

Today’s headsets are able to place a user within a Virtual Reality environment and simulate physical interactions through controllers or the headset itself; any other interaction is typically generated through algorithmic assumptions of where the user’s body should be.

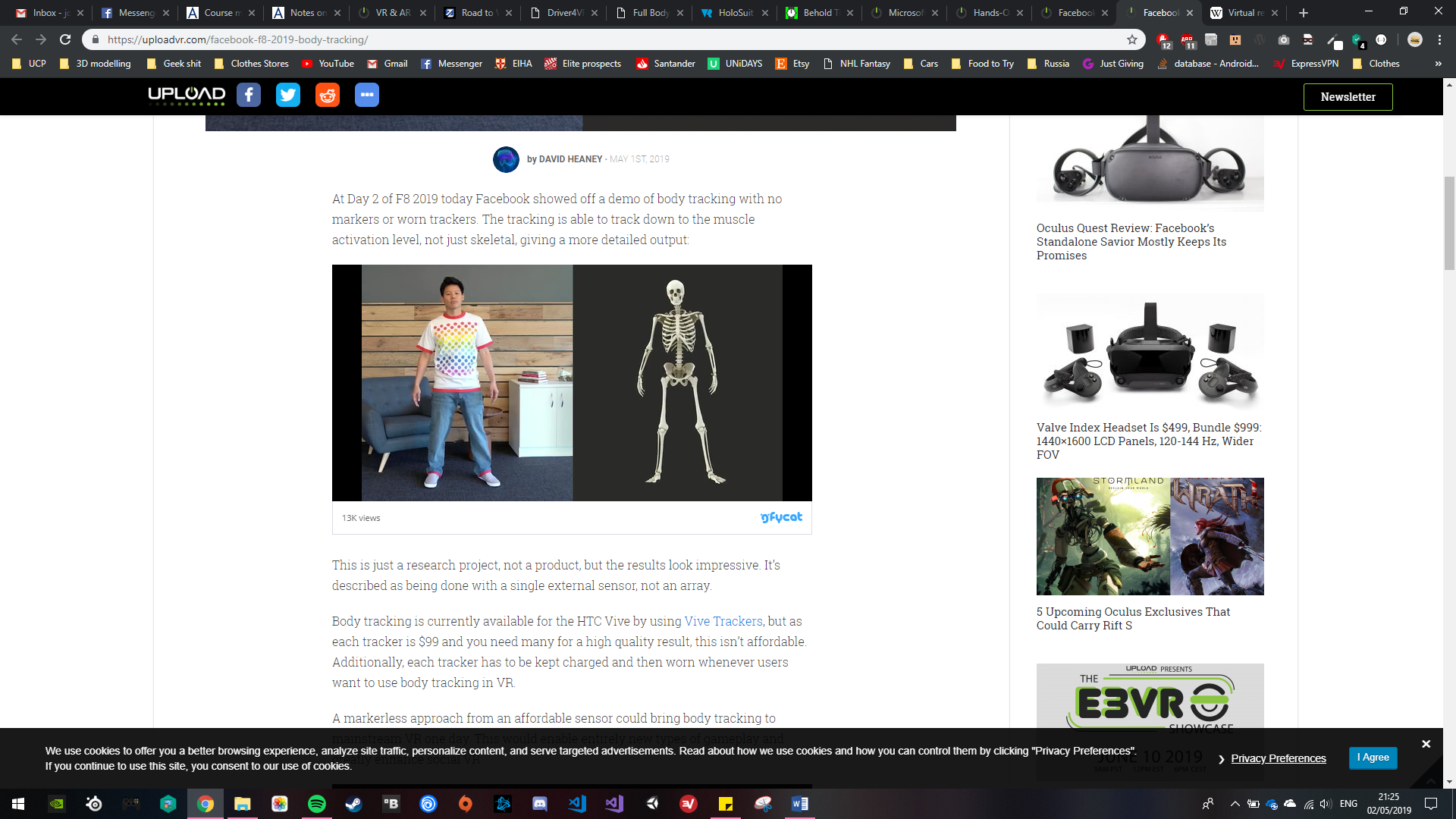
Facebook are currently carrying out a research project, not a product, where they are trying to create a high quality markerless body tracking system from using a single sensor; A snippet of the video they published can be seen to the right, which shows a user being present and the sensor mapping a human skeleton in proportion to where joints and limbs would be in real life, realistically portraying the user in a virtual environment. One of the points UploadVR bring up in this topic is body tracking, and although it is currently available for the HTC Vive, each tracker is $99 and in order to get a high quality mapping result, the user would need to buy multiple sensors and place them all over their body; as UploadVR stated “this isn’t affordable, additionally each tracker has to be kept charged and then worn whenever users want to use body tracking in VR”. *(Uploadvrcom, 2019)*

Figure 1: Facebook Full Body Tracking

With Facebook investing into this project, it pushes forward companies and developers making affordable alternatives which could bring body tracking to mainstream Virtual Reality; “This would enable entirely new types of gameplay and greatly enhance social VR”. *(Uploadvrcom, 2019)*

This opens a path for developers to experiment with new and current technology, to try and find a feasible method for applying full body tracking within the virtual environment. With this, the opportunities expand way beyond the limit they’re currently at. As mentioned above with Facebook’s most recent project is part of their stride towards offering users “photorealistic avatars that looks and moves like the user operating it” and creating a whole new dimension to social networking. *(Rachel metz, 2019)*

## Current Virtual Reality Systems Available

Virtual Reality headsets are one of the top technologies to own in 2019 *(ReferencesTechradarcom, 2019),* a variety of companies have created their own headsets; with some companies are on to their 2nd or 3rd generation of the technology. The VR headsets listed below are headsets which are dependent on a desktop, laptop or console which can deliver high performance and quality - this means excluding independent headsets or ones that require mobile phones to function and which are currently available on the market to purchase.

The top VR headsets currently available in the consumer market:

(no particular order)

**HTC Vive**

Figure 2: HTC Vive System

**Price:** £499

**Design Features:**

* Single front-facing camera, adjustable velcro strap
* Realistic HD haptic feedback controllers
* 1080 x 1200 per eye, with a refresh rate of 90 Hz
* 110-degree field of view

**Tracking Compatibility:**

* Two standard base stations for VR tracking
* 3.5m x 3.5m room-scale stage

**HTC Vive Pro**

Figure 3: HTC Vive Pro System

**Price:** £799

**Design Features:**

* Dual front-facing cameras, adjustable comfort dial
* Free to move around wirelessly with VIVE Wireless Adapter
* 1440 x 1600 per eye, with a refresh rate of 90Hz
* 110-degree field of view

**Tracking Compatibility:**

* Backwards compatibility with original base stations
* 7m x 7m room-scale stage

**Oculus Rift**

Figure 4: Oculus Rift System

**Price:** £349

**Features:**

* Field of view: 100-degrees
* Display: 960 x 1080per eye, AMOLED
* Xbox Integration

**PlayStation VR**

Figure 5: PlayStation VR System

**Price:** £199

**Features:**

* Uses technique called “reprojection” to effectively double FPS from 60 to 120
* 5.7” OLED panel, display resolution of 1080p
* Nine positional LEDs on surface for camera to track 360-degree head movement
* 960 x 1080 per eye

**Samsung Odyssey Windows Mixed Reality Headset**



Figure 6: Samsung Odyssey Windows Mixed Reality System

Price: $399.99

**Features:**

* Resolution: 2880 x 1600
* Refresh rate: Up to 90Hz
* Field of view: Up to 110˚
* Integrated AKG headphones and dual array mics

## Current Full Body Tracking Systems

Below is a list of Full body tracking systems that are currently available in the market today, systems that have been discontinued or are labelled as projects will not be included as the technology isn’t in production for a consumer to purchase.

Each system is unique, and the list won’t include multiple systems from the same manufacturer so that a wider variety of the systems available can be shown and their features listed and explained; a more in-depth breakdown of the Xbox 360 Kinect can be found further down in “Chapter 4 Implementation”.

**Microsoft Xbox 360 Kinect**



Figure 7: Xbox 360 Kinect

**Price:** £10

The Kinect is a motion detecting camera, the technology behind the face is an RGB camera, a multi-array microphone and a depth sensor. These three elements combined allow the Kinect to perform body capture, facial and voice recognition.

**HTC Vive Tracker**

**Price:** £109.99

The Vive tracker is an accessory to one of the HTC Vive Systems; it is a motion tracking sensor and is designed to be attached to physical accessories such as straps and or controllers; the tracker can also be tracked via the same lighthouse system. The Vive tracker looks like a little pod as seen from figure 8 to the right, the technology behind the tracker is essentially the same as the hand controllers with both offering the same tracking capabilities, the same design features by using multiple divots located around the casing for infrared purposes, but the difference being the user cannot interact with the tracker.

Figure 8: HTC Vive Tracker

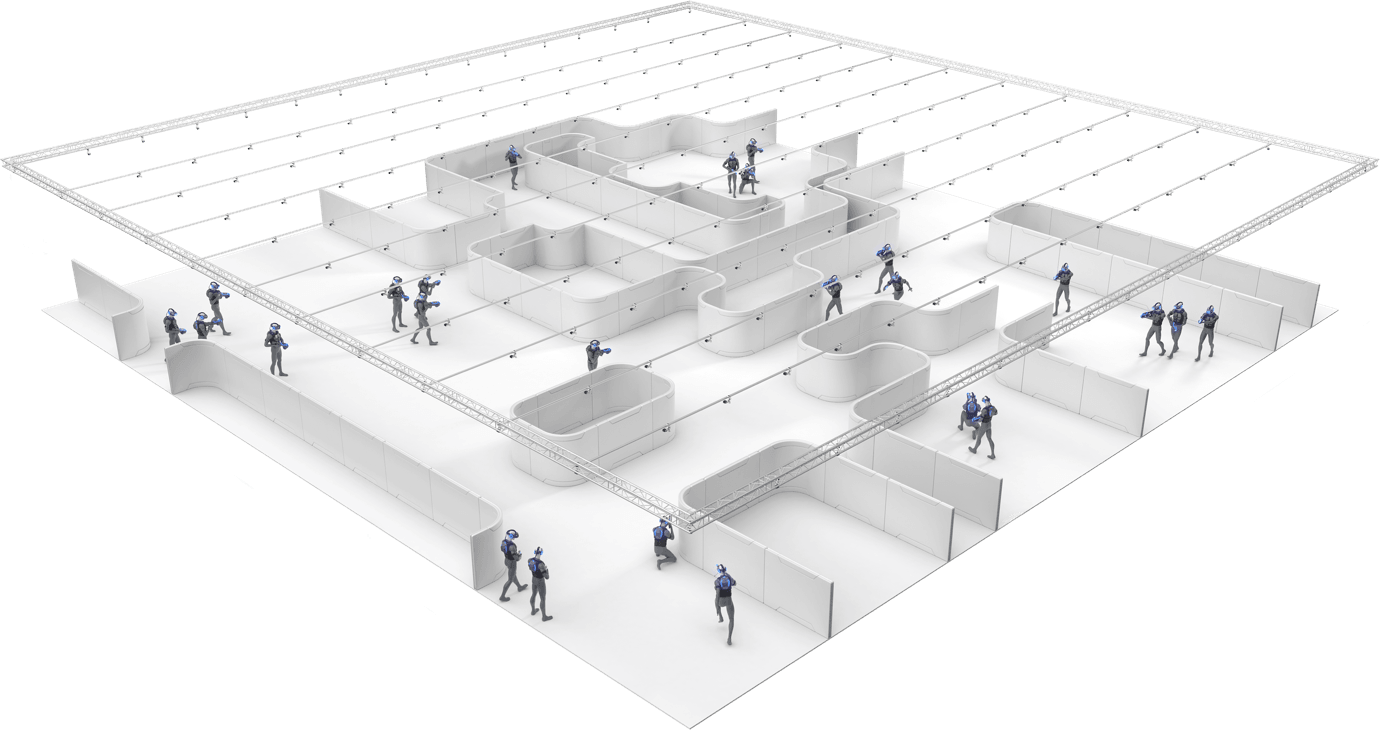
**OptiTrack**

Figure 9: OptiTrack System Layout Diagram

**Price:** Varies depending on package configuration ($8,316 - $500,000+)

Accurately track a room full of HMD’s (head mounted displays) or the whole user’s body with a single tracking system.

This system consists of:

* **LED controller** which is a small PCB controlling eight LEDs – each LED use one for each tracked object. The controller emits a wide-angle invisible IR pulse seen by Slim 13e or other OptiTrack cameras.
* **Base Station** which is typically installed at the center of the tracking area, the purpose of the base station is to keep the LEDs synced with the OptiTrack cameras.
* **Slim 13E Cameras** which are OptiTrack’s 1.3 MP, 240 fps freeze frame tacking cameras.

*(Optitrackcom, 2019)*

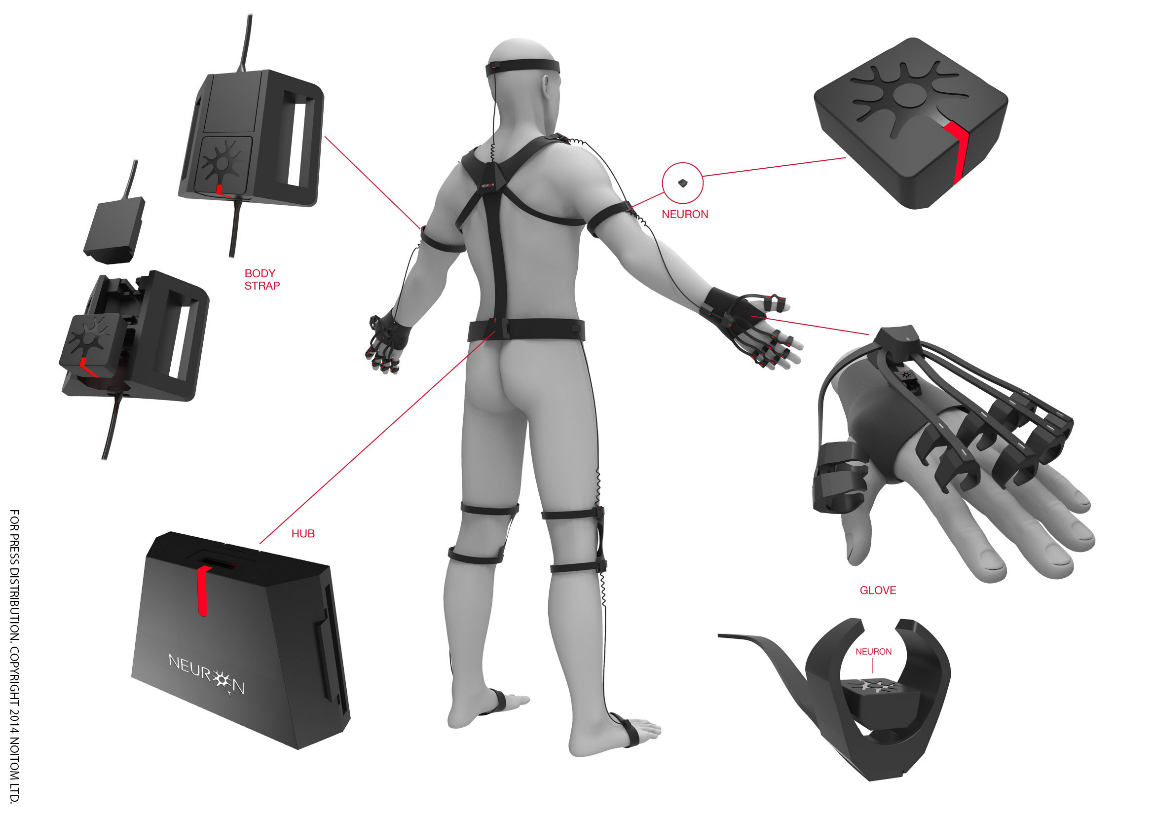
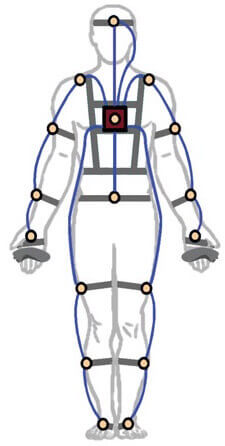
**Perception Neuron**

Figure 10: Perception Neuron Full Body Tracking System

**Price:** £1,799

Perception Neuron is a small, adaptive and versatile motion capture technology. The modular system is based on the NEURON, an IMU (Inertial Measurement Unit) composed of a 3-axis gyroscope, 3-axis accelerometer and 3-axis magnetometer. The strength of the system lies in Perception Neuron's proprietary Embedded Data Fusion, Human Body Dynamics and Physical Engine algorithms which deliver smooth and true motion with minimal latency.

The Perception Neuron 9-Axis sensor units output data at 60fps or 120fps. The data stream is channelled to the HUB where it can then be transferred to a computer in three different ways either via Wi-fi, USB or recorded onboard using the built-in micro-SD slot. *(Neuronmocapcom, 2014)*

**PrioVR**

**Price:** £1,200

PrioVR uses high-performance object detecting sensors to provide 360 degrees of low-latency, real-time motion tracking without the need for cameras, optics or line-of-sight. PrioVR’s sensors are placed on key points of the user’s body to capture their movements. PrioVR is wireless, allows for multiple simultaneous users, and will work anywhere–indoors or out, in capture spaces of any size.

Figure 11: PrioVR System

The PrioVR Dev Kit includes:

* two hand-controllers with action buttons
* triggers
* joysticks

The PrioVR suit comes with demo games so you can experience the freedom of full-body gaming from day one.

*(Yostlabscom, 2019)*

**TVico**

**Price:** $239.99

TVico can be developed with two different setups in mind, it can either be developed as a standalone computer and apps are installed and run on the device, or it can be used as a peripheral and run on Android, Windows or Linux.

Figure 12: TVico VR Sensor

The device offers a range of features and capabilities, to list a few of its major points:

* RGB & Depth Sensor
* Full Body Tracking
* Gesture Recognition
* User masks
* 3D Point Cloud
* Unity and Unreal plugins

*(Tvicoio, 2019)*

## Justification of device for full body tracking

The device chosen to track the user’s body was the Xbox 360 Kinect, one of the main requirements when making this decision was to have a device which didn’t require any external devices or apparatus to be attached to the user’s body; making the flow from turning the device on to using it as simple as possible.

Next requirement was complexity, the aim was to find a device which wasn’t too complex for the user to operate but had the required components to deliver what was required (full body tracking capabilities). The complexity of a device has an influence on the price and one of the aims for this project was to find a cheap alternative to the more expensive products on the market for full body tracking.

Taking all of the above into consideration, the Xbox 360 Kinect was a perfect match for the guidelines set when choosing a device as it offered all that was required for a miniscule fee.

## Current available platforms for 3D Environment development

The platforms that have been aimed at are ones that are well established in the 3D game development industry and have ample online support and documentation, the reasons behind this is because this is the first time doing this type of development on a project. Therefore, there is sufficient support along the way and will ensure development won’t be hindered for too long of a period of time. *(Sundaysundaeco, 2018)*

There are numerous development platforms out there, but there are 2 specifically that are leading the industry with the advanced features they offer and the support that is behind them.

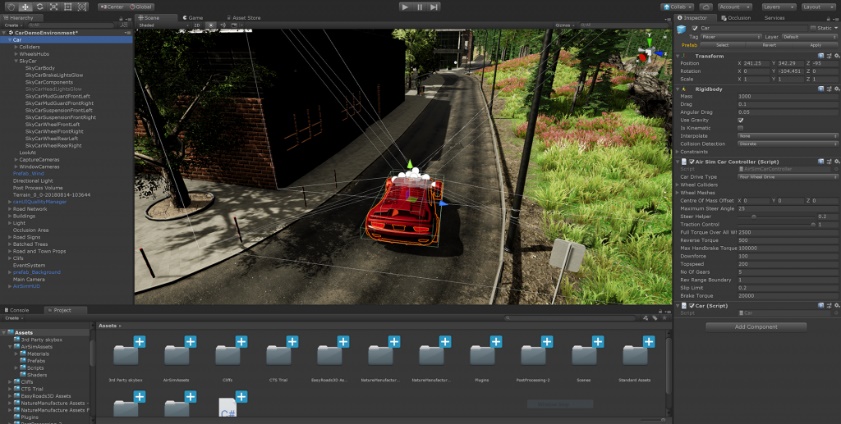
The 2 listed platforms below are:

* Unity
* Unreal Engine

The aims for selecting the right platform for development was to find which one had the better user experience when putting together a simple scene, adding mechanics to objects and which offered the best resources in terms of support and assets.

**Unity**

Figure 13: Unity 3D



**Price:** The software is free for personal use, $25/ month for the Plus edition, $125/ month for Pro edition and prices for enterprises are available upon request.

**Description**

Unity is a cross platform tool developed by Unity Technologies. As of 2018 the engine had been extended to more than 25 platforms; the engine can be used to create 3D, 2D, Virtual Reality or Augmented Reality games, simulations or any other experiences. *(Arstechnicacom, 2019)*

**Programming languages used for development**

Unity uses the following programming languages for development:

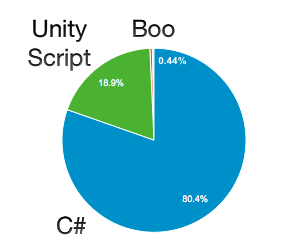


Figure 14: Unity Development Statistics

As you can see from figure above, C# has a very dominant position for programming language used for development with Unity; the chart above was gathered from Unity’s statistics they’ve realised. *(Unity3dcom, 2019)*

**What features does it offer?**

Unity has an extensive list of features; the list below highlights some of the main features which are highly relevant to the project:

All-in-one editor: “Available on Windows, Mac, and Linux, it includes a range of artist-friendly tools for designing immersive experiences and game worlds, as well as a strong suite of developer tools for implementing game logic and high-performance gameplay”. *(Unity3dcom, 2019)*

2D & 3D: “Unity supports both 2D and 3D development with features and functionality for your specific needs across genres”. *(Unity3dcom, 2019)*

Efficient workflows: “Unity Prefabs, which are preconfigured Game Objects, provide you with efficient and flexible workflows that enable you to work confidently, without the worry of making time-consuming errors”. *(Unity3dcom, 2019)*

Physics engines: “Take advantage of Box2D, the new DOTS-based Physics system and NVIDIA PhysX support for highly realistic and high-performance gameplay”. *(Unity3dcom, 2019)*

**Supported Platforms?**

* IOS, Android and Windows Phone
* Windows, Mac and Linux
* PlayStation 4, PlayStation Vita, Xbox One, Nintendo 3DS and Nintendo Switch
* SteamVR, Oculus Rift, Google Cardboard, PlayStation VR, Gear VR and Windows Mixed Reality
* Tizen OS
* Fire OS

**Learning Curve and Support**

“Unity, one of the world’s leading video game development platforms” *(Dailydotcom, 2018)* is an ever-extending platform which is constantly being updated to support the worlds latest programming and game development trends. *(Dailydotcom, 2018)*

With the various online tutorials which cater to any genre of development you wish to partake in, to the documentation listing and explaining every aspect and feature within the software, and to the online blogs and forums which are open 24/7 Unity is a platform where a beginner to the industry and rapidly develop into an expert and become part of the industry they learnt not that long ago.

**Unreal4**

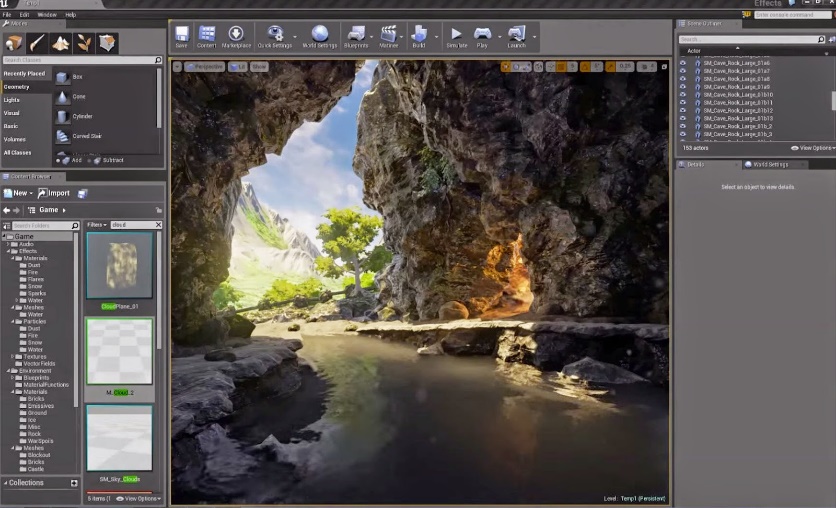


Figure 15: Unreal Engine

**Price:** Free for small projects, 5% royalty on gross revenue above $3,000 per product, per quarter.

**Description**

Unreal Engine is a game engine developed by Epic Games. It Is a complete suite of tools designed to help developers reach their goal; while still being flexible enough to ensure success for teams of any sizes.

**Programming languages used for development**

Unreal engine 4 uses C++, there are plugins available in Unreal’s asset store which can allow for development with other languages. However, not all are officially supported by Unreal.

**What features does it offer?**

Unreal Engine has an extensive list of features; the list below highlights some of the main features which are highly relevant to the project:

Built for VR, AR & MR (XR): “Because of Epic’s close collaboration with global leaders in hardware and software, Unreal Engine provides the highest quality solution for creating virtual reality (VR), augmented reality (AR) and mixed reality (MR) experiences. UE4 is natively integrated with the most popular platforms and offers features like forward rendering, multisample anti-aliasing (MSAA), and instanced stereo rendering”. *(Unrealenginecom, 2019)*

Full Editor in VR Mode: “Reach out, grab and manipulate objects with the power of Unreal Engine at your fingertips. The full Unreal Editor runs in VR with advanced motion controls so that you can build in a “what-you-see-is-what-you-get” environment. It’s the most robust, feature-complete and capable VR development solution in the world”. *(Unrealenginecom, 2019)*

Content Browser: “Use Unreal Engine 4's Content Browser to import, organize, search, tag, filter and modify project assets within the Unreal Editor. Drag and drop assets directly into your scene and build out your world. Create asset collections to be used for individual work or shared with other developers”. *(Unrealenginecom, 2019)*

Limitless Extensibility: “Integrate virtually any technology into your Unreal Engine project by leveraging the modular plugin system. Free source access affords opportunities to create your own implementations for middleware packages, plus there are countless UE4 tool and feature integrations already available through the Marketplace and GitHub community”. *(Unrealenginecom, 2019)*

**Supported Platforms?**

* Windows, Mac and Linux
* SteamOS and HTML 5
* IOS, Android and Windows Phone
* Nintendo Switch, PlayStation 4 and Xbox One
* HTC Vive, Oculus Rift, PlayStation VR, Google Daydream, OSVR and Samsung Gear VR

**Learning Curve and Support**

Unreal Engine, like Unity, is one of the world’s leading video game development platforms. *(Sundaysundaeco, 2018)* However, Unreal take a different approach with ways a developer can build within the platform. Unreal Engine uses the C++ language within the engine. C++ is a mid-level programming language as it binds the gap between a machine level language and high-level languages; for beginners, the learning curve can be a quite steep as this language allows for more control when programming. This can be both beneficial and a disadvantage for not only development time, but time spent learning the platform. *(Edobacomab, 2018)*

Unreal’s online services currently support nearly 250 million developers. *(Epic Gamescom, 2019)* “As a game developer ourselves, we've tackled numerous hard problems over the years. With Epic Online Services, we're sharing the fruits of our labour with the rest of the development community for free.

Epic Online Services will work across any engine, any store, and are designed to be integrated with any major platform. With a single SDK, you can easily access any of the services offered by Epic”. *(Epic Gamescom, 2019)*

## Justification of platform used for development

Prior to tackling this major project, mini projects which focused on individual mechanics which would later be combined would be experimented with in each platform; this provided a better understanding of the process of development and the level of difficulty of implementing said mechanics in each of the engine.

After conducting these mini projects, Unity proved to be the friendliest and one that was easy to develop with, it showed to be more suitable for programming as it has a real-time code refreshing system with a good prefab system.

Although the graphics in Unity didn’t hold up to as high of a standard as they did in Unreal but taking into consideration the end purpose for the platform was to develop a game that would be displayed in a Virtual Reality headset, the graphics weren’t needed to be of high detail.

The winning factor for choosing Unity was the UI was friendly enough for easy development and Unity offered several advanced tools that are not restricted in the personal free version; Unreal has comparatively less collection of tools and add-ons as compared to the Unity engine. *(Capitalnumberscom, 2017)*

# Chapter 3: Methodology

## Overview

The main approach to this project was to gather research from existing computing projects that have had experience with adding functionality like this to a game, and articles on the different techniques of technology behind tracking the body in Virtual Reality; this information will be gathered from resources on the internet such as recorded seminars, from websites with trusted resources and from books.

The ideology behind the process for the project after the research was to develop small protypes of the different areas of functionality the complete game will offer; this allowed for experience in the individual elements of the game and offered a more in-depth understanding of how they work, also affords development once the mini prototypes have been built as they can be built upon and polished to a final product.

The framework that was used to guide this project is Agile. One of the pros of using Agile, mentioned by Laury Hales, currently doing her PHD and Director and European Operations at Decypher is “Agile is more flexible, adapting to changes much more rapidly than other project management approaches”. (Study.com, 2018)

Testing of the prototype will be carried out by the researcher without the need to involve other participants. The hardware required for gathering the motion data was camera based (such as the Microsoft Kinnect) and did not require any attachment to the body to function.

The results from this testing can be used to compare against the aims set at the beginning of the project. This will determine whether or not the project was successful, a failure and outline specific areas which could be improved upon to further better the results in the future.

## Proof of Methodology

**Meetings** – A list of meeting minutes can be found in the appendix under “Section 1 Part A: List of Supervisor Minutes”

**Gantt** **Chart** – Two images can be found in the appendix under “Section 1 Part B: Gant Charts”; here can be seen a prior planned Gantt chart which was done at the beginning of the project, and underneath an updated Gantt chart with actual dates and time taken for completion of said tasks.

# Chapter 4: Implementation

## Lower Body Tracking

The technology behind tracking the user’s lower body is the Xbox 360 Kinect V1 from Microsoft, this device makes use of a depth sensor which consists of an infrared projector and a sensor, the projector projects a continuous infrared pattern over its field of vision; which is how the sensor interprets the environment and bodies that are within its vision.

Figure 16: Xbox 360 Kinect V1

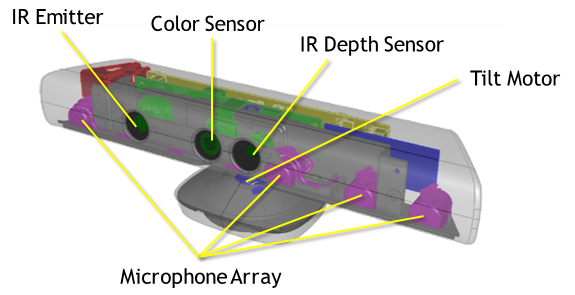
The Kinect has a 30fps camera with a min to max distance of 40 cm in near mode and up to 4.5m in regular state; its horizontal field of view is 57 degrees and the vertical field of view being 43 degrees; with this range of vision the sensor can detect up to 20 joints (this translates to 2 full skeletons tracked); a diagram of the Kinect V1 can be seen in figure 17.

Figure 17: Diagram of the Kinect V1

The Kinect has 3 versions, Kinect for the Xbox 360, released in 2010 and supporting the first version of the Kinect software “V1”, next was the Kinect for Windows, which was released in 2012 and supported the same version as the Kinect for Xbox 360 at first but was later updated with the next generation of Kinect that was to come out.

The latest and last version released by Microsoft was for the Xbox One , which was released in 2013 and supported the new and improved “V2” version.

Since the version of Kinect used in this project was for the Xbox 360, it wasn’t compatible for Windows 10. However, with the aid of 3rd party software (Driver4VR) the Kinect is able to connect to Desktops and function as normal.

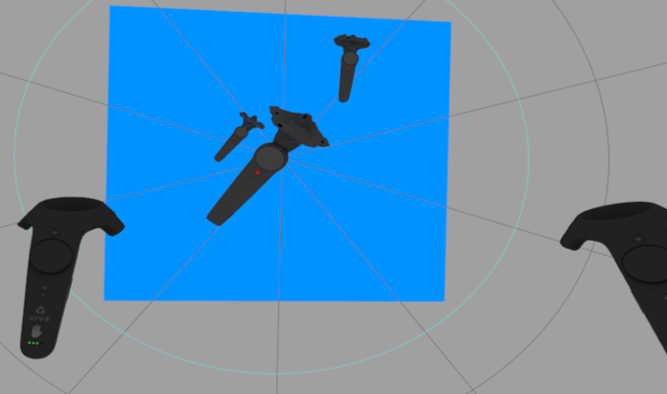
Driver4VR emulates Vive Trackers via the Kinect’s motion sensing data, this provides a simplistic mapped skeleton of the user’s body which you can see an example from figure 18 below; with specific joints being pinpointed for the application of the 3 sensors attached to the user’s body when implemented into a virtual reality environment (this will be shown and explained below).

A screenshot of a video game

Description automatically generated

Figure 18: Driver4VR Mapping Skeleton to User

With the software calibrated to the user’s body, it then applies 3 sensors which mimic the HTC Vive trackers; two being placed on the user’s feet (one on each foot) and one placed in the centre of the user’s hips.

The result of this is that 3 additional sensors are applied along with the two controllers from the Vive, forming a precise map of where the user’s body is and all of their limbs, as seen from figure 5 below.

Right hand

Left hand

Left foot

Hips

Right foot

Figure 19: 5 Sensors (head, hands, hips and feet)

## Tracking the hands and head

The Virtual Reality headset chosen for this project was the HTC Vive, developed by HTC and Valve Corporation, this system offers “room scale” tracking technology allowing a user to move in the environment around them and interact with it by the use of motion-tracked handheld controllers.

The list of basic components offered by this system are as followed:

**Vive Headset** – a 90 Hz device with a 110-degree field of view. The device uses two OLED panels, one per eye, with each offering a display resolution of 1080x1200 (a combined total of 2160x1200 pixels). Safety features on the headset include a front-facing camera, this allows the user to see their surrounding environment without removing the headset; as seen from figure 20, the headset has multiple divots located around the case which, like the controllers, use this for the infrared sensors to enable detection from the base stations via infrared pulses to determine the location of the device within the environment. *(Sean buckley, 2019)*

Figure 20: HTC Vive Headset

Other sensors in the headset include a G-sensor(accelerometer), gyroscope and proximity sensor.

**Vive Controllers** – Consisting of multiple input methods including a track pad, grip buttons and a dual-stage trigger. Across the ring of the controller are 24 infrared sensors that detect the base stations to determine the location of itself to within a fraction of a millimetre (updating 250Hz to 1KHz). *(Vrheadscom, 2016)*

Figure 21: HTC Vive Controllers

With the lighthouses projecting infrared signals, an array of sensors on the controllers can accurately track their location in relation to the stations; this allows the system to provide prevision within millimetres, providing a much of an immersive experience as possible. *(Ben lang, 2017)*

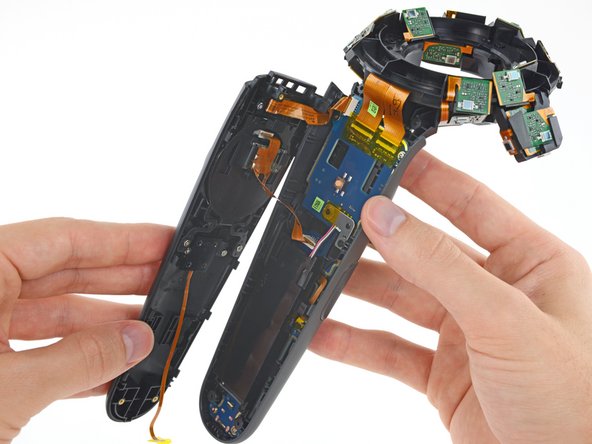
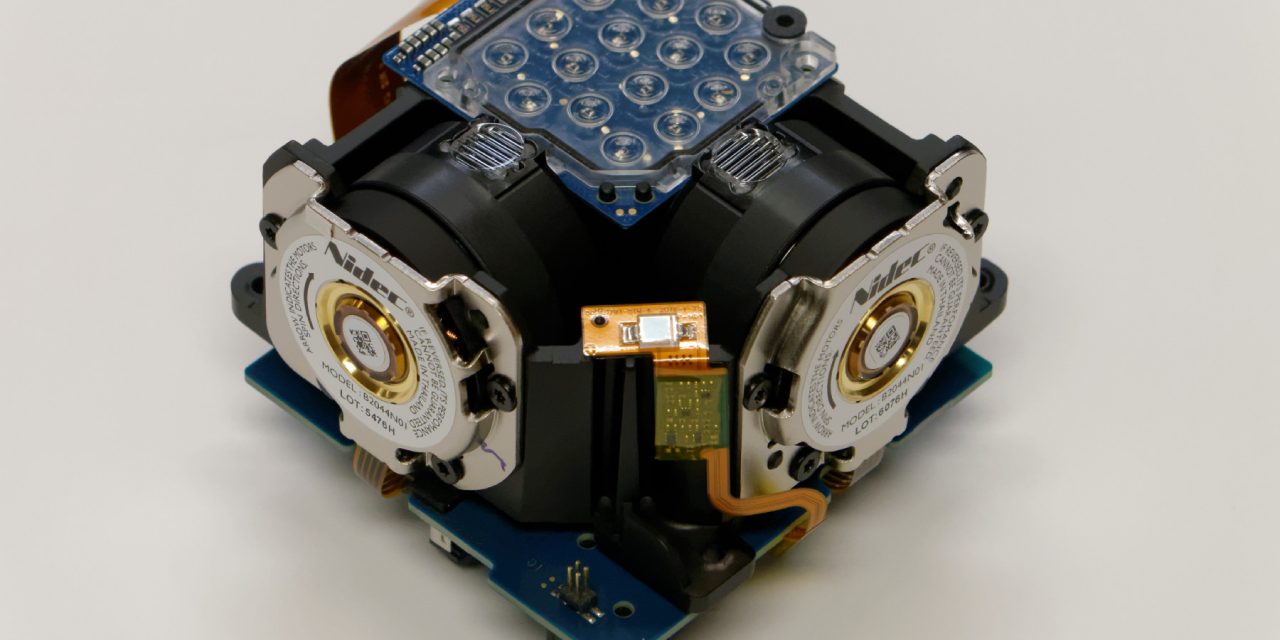
As well as the array of sensors mentioned above, the controller is also covered in infrared sensing cavities, each dip retains an infrared filter, which attempts to strip down incoming light to the guiding beams emitted by the HTC Vive's base stations. *(Vrheadscom, 2016)*

Figure 22: HTC Vive Controller without casing

**Vive Base Stations** (Lighthouses) – These are, at minimum, two blocks which create a 360-degree virtual space by emitting infrared pulses at 60 pulses per second *(Ben lang, 2017)* which are picked up by the devices operating within that area.

Figure 23: HTC Vive Base Stations

These lighthouses are placed at opposite corners of the room, preferably at a height higher than where the user’s head would be; these lighthouses do not communicate with the PC and they are not sensors.

The theory behind the tracking is the lighthouses flood the room with non-visible light, by emitting wide-angle two-dimensional IR laser beams across the entire room the lighthouses pose as a reference point for any positional tracking device to figure out where it is in the real-world environment; this process is done 1 axis at a time, so left-right then top-bottom, repeatedly - before each sweep they emit a powerful IR flash of light. *(Ben lang, 2017)*

The technology behind the Lighthouses is explained in David Heaney’s article, an author at UploadVR. “Each tracked device contains an array of IR photodiodes (converts light into an electrical current) connected to a chip. This chip measures the time between the IR flash and being hit by the laser sweep for each axis. From this it can determine its position in the room”; a lighthouse’s open structure can be seen in figure 6. *(Uploadvrcom, 2019)*

Figure 24: HTC Vive Lighthouse without casing

## Goalie Equipment

**Catcher** - When applying texture maps to a model, the aim for their application is to enhance the model’s visual appearance (giving it more detail) without the need to physically change its structure; typically, when constructing a 3D model one of the good practices before exporting to an external texturing program is to ensure all of the maps are correct on the model, preventing any unexpected blemishes on the textures when they’re applied to the model.

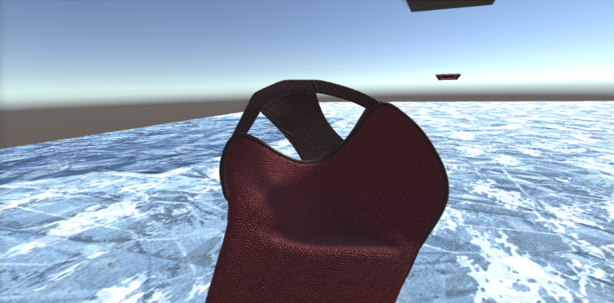


Figure 25: Goalie Equipment - Catcher

As you can see from figure 16, when the glove is in one direction, there is a bridge between the palm of the gloves and the outer stitching arch. However, if the glove is rotated you’ll notice this bridge disappears. This is a little trick that can be done by rotating individual vertices within the 3D modelling software to flip the normals, enabling faces to disappear on one side. The purpose of this was to make this piece of equipment a little easier to use due to it occasionally blocking the user’s vision when making a save.

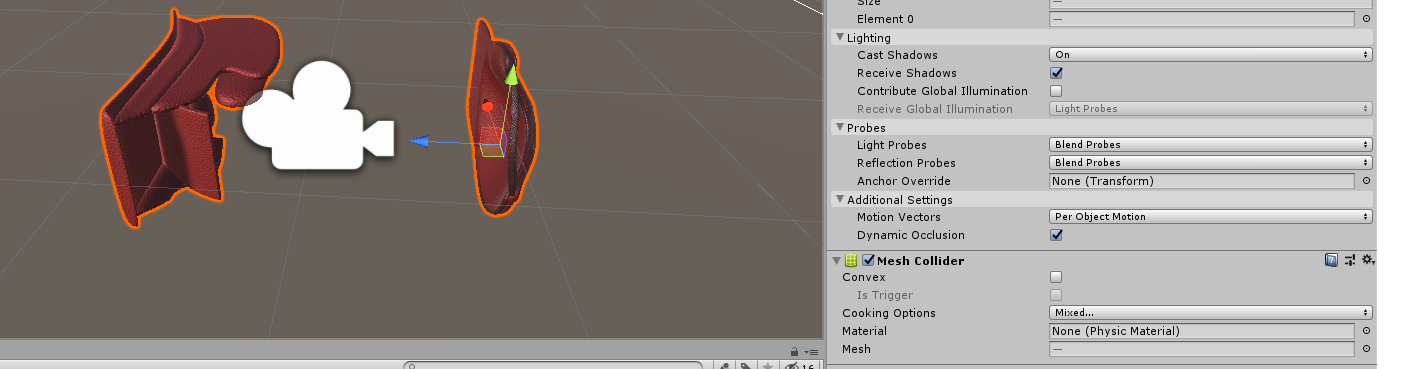
The other pieces of equipment are a blocker, body armour and a helmet; all are basic models with regular texturing with no alterations needed for the user’s benefit.

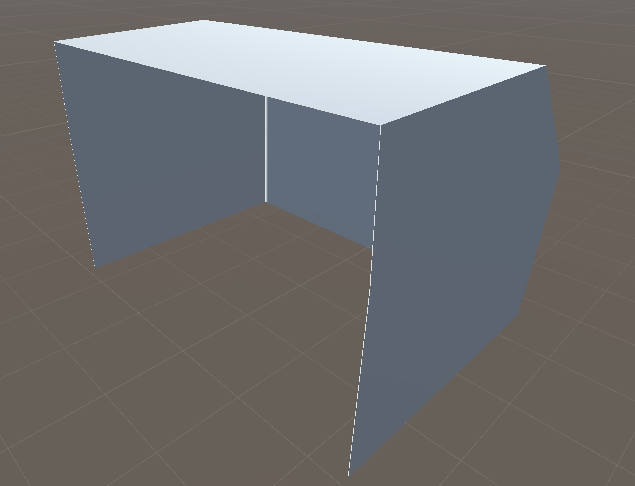
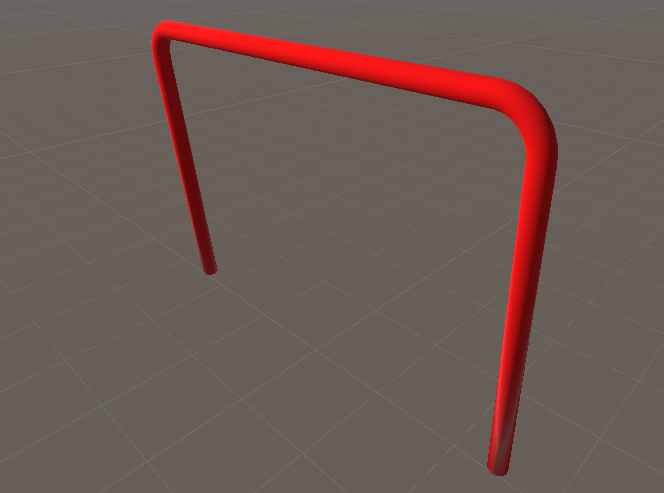
Figure 26: Mesh Colliders

For collision on the equipment, the aim was to ensure all of the equipment responded how the user would expect; this is especially important due to the purpose of the equipment - it’s used for stopping a puck from going into the goal. Therefore, the equipment should feel as an extension of the user’s body.

To meet that aim, each piece of equipment has a mesh collider applied to it; this type of collider takes a mesh asset and builds its collider based on that mesh; this type of collider is far more accurate for collision detection than using primitive shapes. *(Roguecodecoza, 2019)*

**Goal -** The goal is split into two sections, the first being the posts and the other being the netting as seen from figure 18.

Figure 27: Construction of the goal



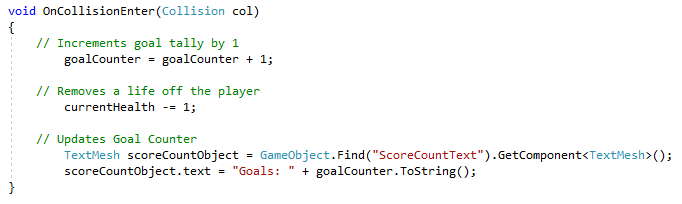
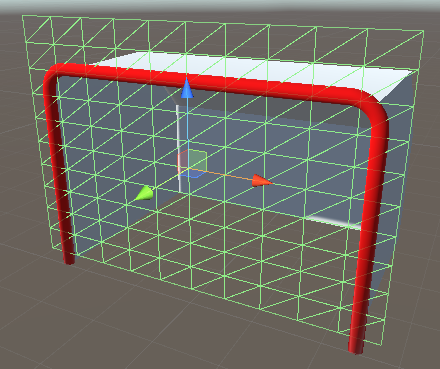
The purpose for having the goal split into 2 pieces is for collision purposes, using the data generated from each object when a puck collides with either. To explain further, if the puck was to collide with the post and not collide with the netting, the program would determine that as no goal and update the score board appropriately.

Figure 28: Sample code of goal detecting script

As seen from figure 19 above, the “onCollisionEnter” function is applied to the goal netting, this function is called when collider or rigid body has begun touching another collider or rigid body.

The reason for using this function and not a “OnTriggerEnter” is because the “OnCollisionEnter” is passed the whole Collision class which contains information about the contact points of the object and its velocity; which is useful because these are the values which will be used to add force to the object and add realistic behaviour to itself as it travels from point A to B.

**Puck Shooter** - The puck shooter has 3 stages to its process: gathering random coordinates within the defined shooting area, generating the velocity of the puck and firing the puck at the gathered coordinates with the defined velocity.

In order for the puck shooter to gather coordinates, a firing area needed to be defined rather than trying to scale the entire world space.

A single plane has been placed on the face of the goal with the mesh renderer turned off, which enables the plane to be invisible to the environment, but its mesh remains which can later be used for calculations, while the next figure is of the script behind the Puck shooters “GetTarget” feature.

Figure 29: Shooting area

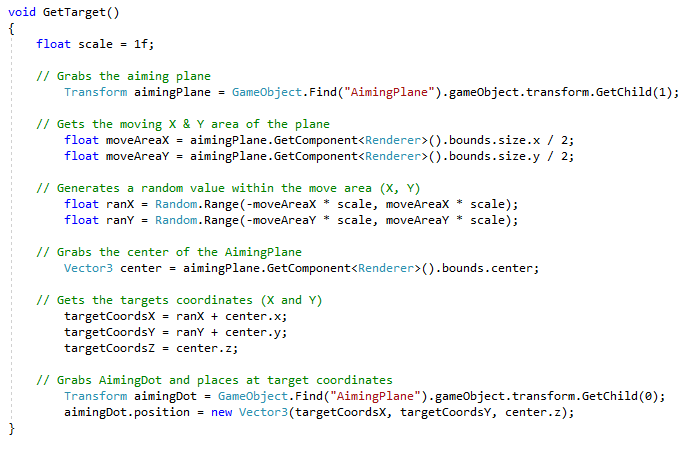


Figure 30: Sample code for gathering target

To explain figure 21 further, the method above grabs the plane that was placed across the face of the goal, gets both its X and Y area by taking its boundaries and dividing them by 2.

With these values a random X and Y coordinate can be generated, by taking both X and Y positive and negative moving areas (positive and negative because the centre value would be 0) and multiplying them by the scale value. This allows the Puck Shooter access to the whole plane; with these values the program can take them and place them into variables which will hole the generated X, Y and Z coordinates which will be used for the target position within the net.

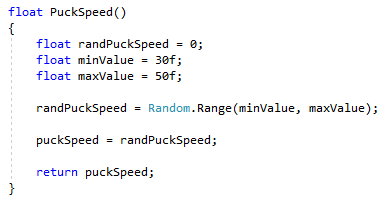
Now the shooter has coordinates to shoot at, the next stage is to calculate the velocity the puck will travel at.

Figure 31: Generating Puck Velocity

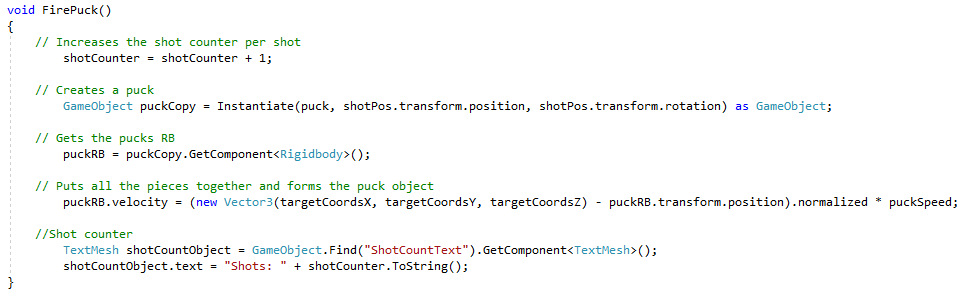
As seen from figure 22, a puck has a minimum and maximum value with which these values are plugged into the parameters of the “Random.Range()” method; that method then returns a value which is placed onto the current puck being shot at the goal.

Figure 32: Firing the puck

The last stage is to take all of the values generated and place them into a puck which will be fired at the goal. From the top of figure 23, a shot counter is incremented each time and this value will later be used for the score board, from there a copy of the original puck prefab is made, the program then grabs the pucks rigid body which allows physics to be applied to it, said physics being the velocity. The velocity of the puck is calculated by taking the target point’s X, Y and Z values and subtracting the pucks current position, normalising that value and then multiplying it by what speed that puck will be traveling at.

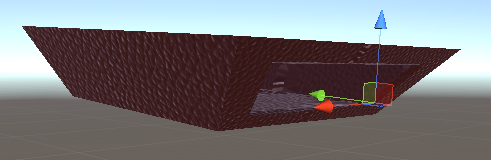
The current position of the puck is its spawning point which has been defined at the end of the puck shooters barrel as seen from figure 24 below.

Figure 33: Shooting point

The formula above (targets position – pucks position normalised and then multiplied by speed) is used to get the direction the shooter will need to fire the puck in order to reach its target.

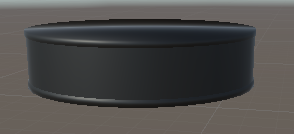
**Puck** - The puck object itself has a mesh collider applied to allow precise collision detection and a rigid body attach to it as well; a rigid body allows an object to receive forces and torque to make it move in a realistic way. *(Unity technologies, 2019)*

Figure 34: The puck

 **Score Board** - The scoreboard consists of the following details seen from figure 26 to the right: the high score, this is incremented every second the user manages to carry on within the game, the amount of shots the user has had taken at them, the amount of shots that have turned into goals and the current number of lives they have left.

Figure 36: Losing lives



Figure 35: Scoreboard

When the game starts the user has 10 lives, every time a puck goes into the goal (scores a goal) a life is taken away. As you can see from figure 26, the user had let in 4 goals and therefore 4 lives have been taken away.

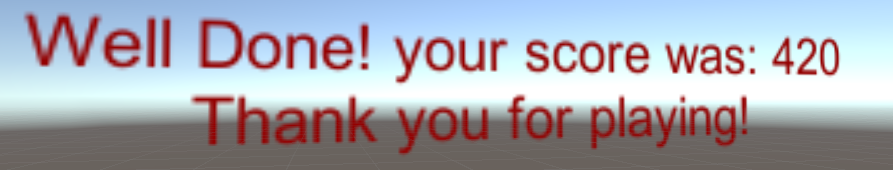
If the user reaches 0 lives, then the game is over, and a message is displayed in front of the user congratulating them for getting as far as they did and displaying their high score as seen from figure 27 below.

Figure 36: End of game message

# Chapter 5: Findings & Discussion

## Findings

In autumn 2017, Microsoft announced they plan to discontinue the Kinect sensor. Early in 2018, on January 2nd, Microsoft announced they will be discontinuing the USB adapter for the Kinect sensor as well.

Microsoft had released 3 versions of the Kinect, each version is shown and explained above in the Implementation section under “Lower Body Tracking”, learning about the discontinuing of the Kinect allowed the understanding that current Kinect drivers and plugins may not work with current versions of mainstream operating systems, such as Windows and software platforms like Unity; this was confirmed later on in the project as the Xbox 360 Kinect, which runs the first version of Kinect, isn’t supported by Unity 5 nor is it supported by Windows 10, stopping the Kinect from being operational in Kinect Studios.

However, the latest version of the Kinect, which is available on the Kinect for Windows and the one for Xbox One, is compatible with Kinect Studios, Unity and Windows 10 as developers have released drivers and plugins which can be installed with ease.

Although importing full body tracking with the Kinect V1 wasn’t possible in Unity, Virtual Reality was to a great extent supported in Unity 5. With the help from software published by a mainstream provider, a Virtual Reality library (SteamVR) was imported into the project where an already structured camera rig could be placed into a scene; from there, the developer can adapt, customise or completely alter how the user can view, interact, manipulate and navigate in the environment.

If the game was run with the camera rig left in its original state, the user would be able to look around the environment but would not be able to interact with any objects in the scene or use any of the controller features they may be accustomed to.

With the library imported, some basic scripts were available which offered simple menu buttons, or navigation tools such as a laser pointer.

However, not all of the scripts needed were available in the library. Therefore, researching further into how C# can be used in Unity, certain features and new syntax was found; for example, the component “Quaternions” was experimented with when making a cannon tutorial in Unity as prior research (Quaternions are used to represent all rotations of an object). *(Unity technologies, 2019)*

Some differences noticed between using C# regularly and using C# in Unity was that while you are able to use Visual Studio for editing and scanning for compile-time errors, to build and run the program that has to be done in the Unity IDE.

Although Unity does use C#, Unity is built on Mono which is an open-source implementation of .Net. Therefore, there are some slight difference relating to that. *(Unity technologies, 2019)*

The workflow with C# isn’t a new concept, previous experience helped the process of programming and scripting within Unity be a smooth process. However, navigating and using Unity’s UI was a new experience, the same as modelling 3D objects in external software for Unity and then importing them into Unity.

When modelling externally and then importing the model into Unity, there are a few factors which need to be taken into consideration. The position of the 3D models coordinates in the external software have an effect on the models position and orientation when imported into a Unity scene, the same is said about scaling; the scaling of an object can be misinterpreted when in the 3D editor environment compared to when it is imported into a Unity scene due to the units of both software’s being different (Unity by default uses meters, Blender by default is in Blender units). *(Unitycom, 2019)*

Moving on to findings with the Vive Virtual Reality System, it was discovered that the Base Stations in a confined room with minimum glass had no issues with syncing and tracking all of the devices within its view; and is an ideal environment for Virtual Reality use. However, the system was originally set up in a large open space with large planes of glass forming some of the walls, being in this environment the system developed bugs and had issues with syncing the base stations and tracking the devices in its field.

## Discussions

Project management is a complex process, while the stages may seem minimal in terms of quantity, once them stages are broken down into tasks, the process can become quite clustered.

When originally planning out this project, the time allocations specified for some tasks proved to be too much and the task was completed with plenty of time to spare. However, with other tasks, the development process went way over the time allocated for it which had an effect of when other tasks could have been started; The Gantt chart reflects this as the original chart compared to the updated one shows considerably more time spent on certain tasks while others show a shorter time spent on them.

Moving onto more specific details within the tasks, as stated previously, Kinect technology had been discontinued by Microsoft and has been out of production for more than 4 years. However, the Kinect is still being used by developers for various other purposes other than for the Xbox consoles; as you can see from figure 37. (Hackadayio, 2019)

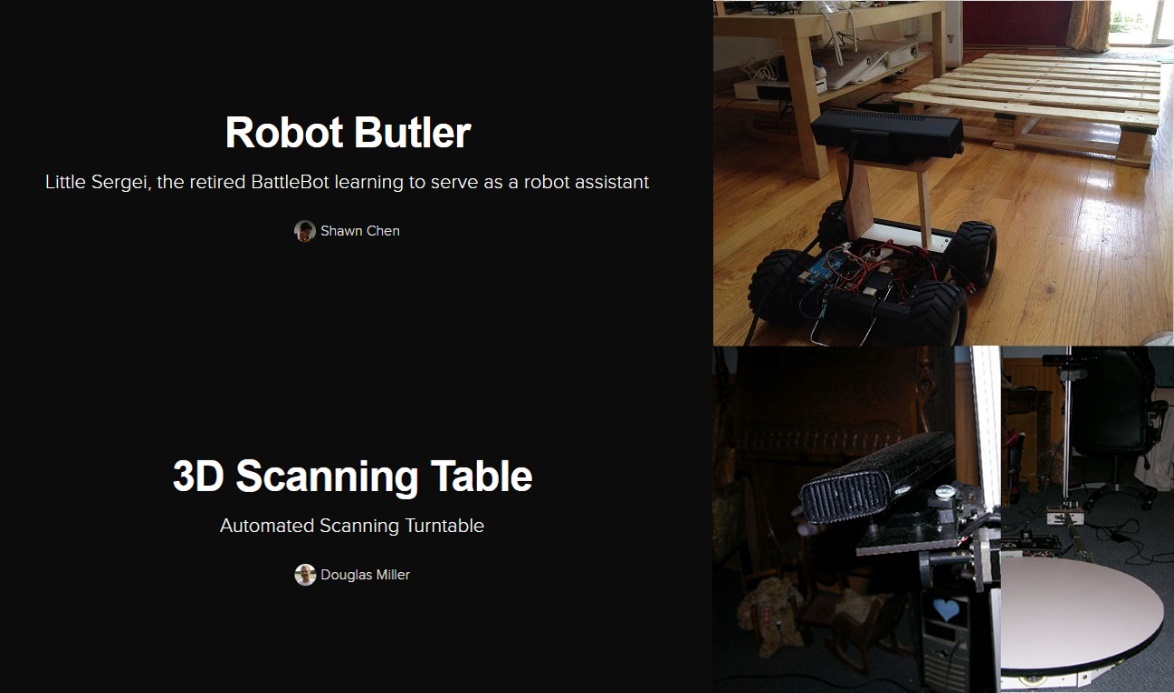


Figure 37: Kinect Projects

The depth sensor technology behind the Kinect is still quite advanced compared to the complexity of full body tracking systems today; comparing it to the latest Kinect released, the V2, there are some obvious performance differences with the V2. The colour camera on the V2 supports a 1920 x 1080 resolution at 30 fps while the V1’s resolution only goes up to 640 x 480 at 30 fps. *(Skarredghost, 2016)*

Below is figure 38 and 39 which compares the visual input between the Kinect for the Xbox360 and the Kinect for Xbox One.

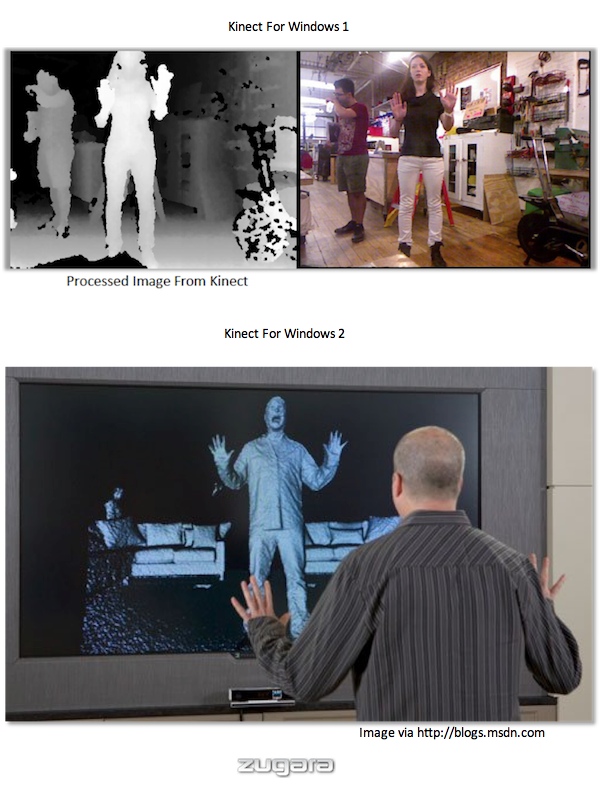


Figure 38: Kinect V1 for Windows

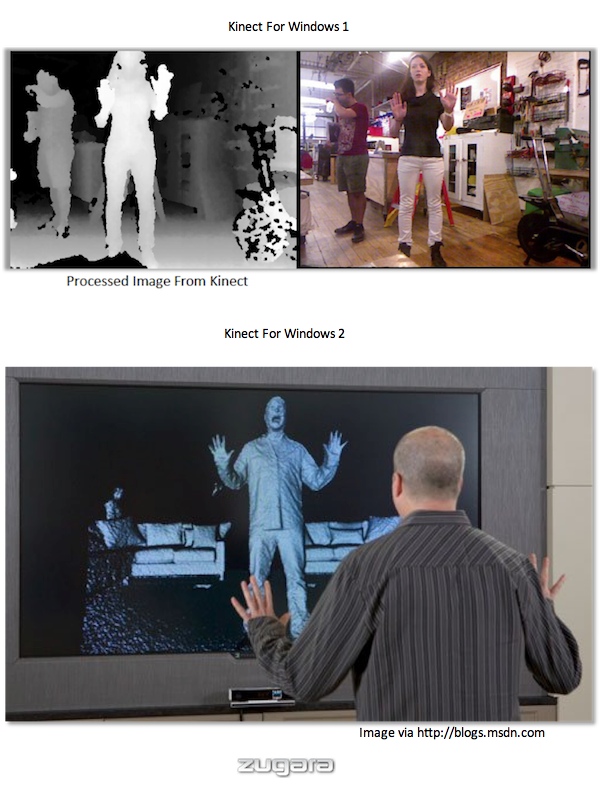


Figure 39: Kinect V2 for Windows

The Kinect for the Xbox One clearly shows it has taken more detail in when capturing the environment, thus the output is by far greater for accuracy; this is due to the higher resolution of the camera it uses. However, for basic body tracking and getting a silhouette of a user the first generation of Kinect holds up quite well and can still serve a purpose.

# Chapter 6: Analysis

Below is listed the outcomes originally set for this project, under each point will the summary on if each outcome was met and or what stopped it from being successful:

* Develop an application with a consistent rate of 90 frames per second

During the later stages of development of the project, this aim was partially met. However, after certain plugins updating, and bugs developing within the scripts of the game, the game’s fps lowered. However, it did maintain on average above 60 fps.

* The finished application will accurately track the position of both upper and lower sections of the user’s body

This outcome was met, evidence of this can be seen from the implementation section under “Lower Body Tracking” and “Tracking the Hands and Head”.

Although this outcome was met, the full implementation of it into the game was not successful due to compatibility issues with the Kinect and Unity.

* When a puck is generated, the puck shooter will select a position within the defined shooting area and fires the puck at the corresponding coordinates

This outcome was also successful, with evidence of this aim being met in the Implementation section under “Puck Shooter”.

* The application will offer collision detection between the puck and the user’s body, so that the user can block the path of the puck stopping it from going into the net

Feedback from user’s who had tried the game confirmed they were able to block pucks that were within their reach. Evidence of implementing mesh colliders and rigid bodies to objects can be found in the implementation section under “Goalie Equipment” and “Puck”.

# Chapter 7: Conclusion & Future Work

## Conclusion

Overall this project was very informing; the experience gained from implementing features such as full body tracking by using technology, which is no longer supported, has afforded not just a better understanding of current technology trends by seeing what other developers have done with the tech, but also the researching skills needed to find alternative methods to get the data required for the result needed.

This project has also given an insight to how a developer can implement a Virtual Reality game into a 3D environment and make the necessary modifications to enhance the users experience and flow of the game a lot smoother; during the prior research period for this project, features such as applying a laser pointer to the user’s controller to allow them to easily track where controller was pointing was implemented just to get a better understanding of how the controllers can interact within the environment.

Even though all of the objectives set were not fully met, this project has provided a clear understanding of how this type of study can be taken further and the different ways it can be implemented to meet specific requirements; this will be further talked about within the “Future work” section.

## Future Work

For future projects, there are definitely areas which will have a different approach. The overall management of the project will be altered so that it gives the research period of the project a substantially longer amount of time; therefore, minimising the possibility of any surprises with incompatibility issues between hardware and software.

As of May 2019, a plugin has been developed which allows for the original Kinect for the Xbox 360 to be compatible with Windows 10.

However, the balance between price and compatibility is clearly shown as a linear line; the more spent on the device, then the more support it has behind it with more frequent driver and plugin updates. The Kinect for the Xbox One is still cheaper than other devices in the market but has just as much support, if not more due to the tech being discontinued, allowing developers to create their own scripts for tweaking how the hardware behaves.

# Appendix

## Section 1 Part A: List of Supervisor Minutes

**11/10/2018:** Scheduled the first meeting with my supervisor for my project. For the first meeting I plan on putting my current situation across; what documentation I’ve completed already, what documentation I’m current working on, and what is left to do.

I’ll put forward some ideas about the game to my supervisor and get input from them to make sure I’m heading in the right direction.

**12/10/2018:** The meeting with my supervisor consisted of me presenting where I’m currently at with my ethics documentation, getting advice on how I should construct my wording in certain areas of documentation, and what is to be expected of me when I’ve completed the documentation and ready to present it to the ethics committee.

The next meeting will take place on the 16/10/2018 where I hope to have completed all documentation ready to present to my supervisor.

**16/10/2018:** A short meeting with my supervisor, from our side it’s thought that my project may not get the approval from the ethics board due to me being a computing student and theoretically not having any previous experience with psychological testing. Therefore, the new aim for the project is to develop a system with offers full body tracking by using the Vive headset and controller along with the Kinect.

**23/10/2018:** A short meeting with my supervisor, all this meeting concluded was a little guidance on areas of the project which could cause me some issues and how I could go of avoiding them. I had done some work on the documentation; which he read over and gave some feedback – I’m on the right path and just need to continue where I left off.

**25/10/2018:** The meeting today with my supervisor gave me a big step forward I the project as I finally got the Kinect, this will allow me to go forward with full body tracking development with the headset and controllers.

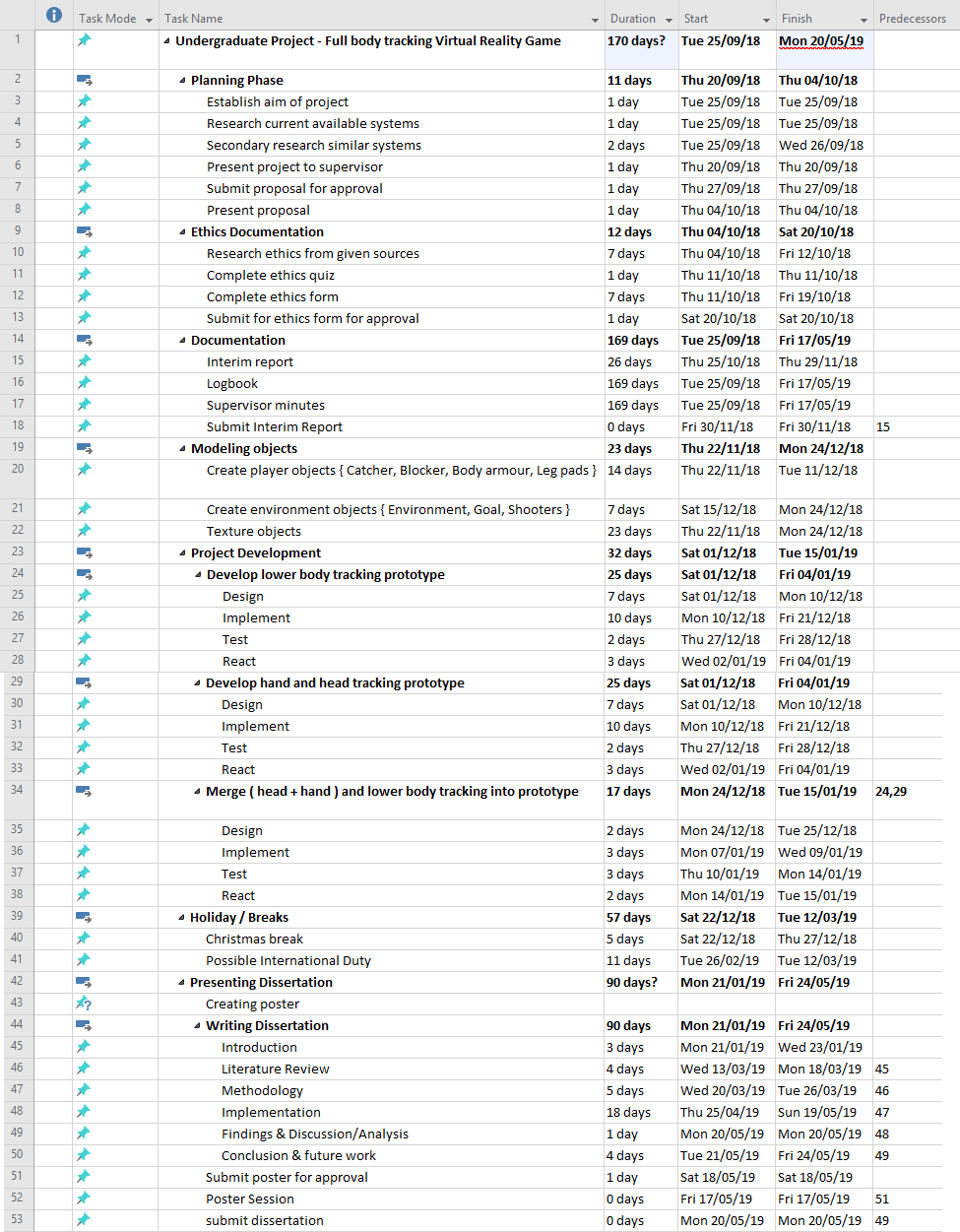
My supervisor read over the documentation I had already done and gave some feedback and ideas for how I could structure it. I also showed him some of the additional stuff like 3d models which will be included in the game.

**13/11/2018:** The meeting today was short and to the point, the aim of the meeting was for me to present what I had done so far with the Gantt chart; to make sure I had not forgot any important tasks or specific time marks I should’ve reached. My supervisor suggested adding some additional break down of tasks to make the project plan more agile, but other than that I had completed the list – all that is required is for me to finish setting dates and then the Gantt chart can be added to the interim report.

## Section 1 Part B: Gantt Chart (Before and After)

**Before**

**After**

****

## Section 2: Articles

**Article 1:** Tom’s Hardware*(Tom's Hardware. 2018)*

Tom’s Hardware is an online publication owned by Purch Group and is focused on technology. It provides articles and the latest news on the most modern technology and hardware available.

The article offers a step by step process of a cheaper alternative for full body tracking in Virtual Reality, it explains the utilities and drivers the user would use and download in order to get the Kinect compatible with the computer and collect data from it.

A weakness is that the article doesn’t address how the user would use and interpret the data they’ve just gathered; therefore, further research on external sources would have to carried out.

The project proposed is developing a Virtual Reality experience by using the HTC Vive headset and controllers with the Xbox Kinnect, to offer full body tracking and full interactivity within the environment.

**Article 2:** VR Focus*(VRFocus. 2018.)*

Kaaya Tech aim to kickstart technology for Virtual Reality usage by trying to develop an affordable HoloSuit; a full body tracking, bi-directional, wireless solution complete with haptic feedback.

The aim for this technology would revolutionise interactivity in a Virtual Reality environment as the suit hopes to offer not just movement of the body but also the movement of the hands, fingers and even the head all while providing feedback to the wear.

The weakness of this research is the equipment isn’t at full market standard yet (totally affordable just yet). Currently, customers can purchase a 1 finger HoloSuit glove for $99, a full 5 finger glove costs $249 and the price for a full HoloSuit is $999+ depending on the number of sensors and haptic devices you prefer.

The project is still in the development phase. This research will influence the project in order to develop a cheaper alternative for full body tracking.

## References

Kirill karev. 2018. Hacker Noon. [Online]. [21 April 2019]. Available from: https://hackernoon.com/behold-the-next-generation-vr-technology-part-2-full-body-tracking-1210ab27ebe4

Rachel metz, cnn business. 2019. CNN. [Online]. [2 May 2019]. Available from: https://www.cnn.com/2019/05/01/tech/facebook-vr-avatar/index.html

ReferencesTechradarcom. 2019. TechRadar. [Online]. [3 May 2019]. Available from: https://www.techradar.com/news/best-gadgets

Study.com. 2018. Pros & Cons of Agile Development Methods | Study.com. [ONLINE] Available at: https://study.com/academy/lesson/pros-cons-of-agile-development-methods.html. [Accessed 30 November 2018].

Tom's Hardware. 2018. You Can Use A Kinect For Full Body Tracking In SteamVR; Here’s How. [ONLINE] Available at: https://www.tomshardware.com/news/driver4vr-kinect-full-body-vr-tracking,35476.html. [Accessed 01 November 2018].

UploadVR. 2019. HTC Will Open Source Full-Body Tracking For Vive With Tracker. [ONLINE] Available at: https://uploadvr.com/htc-will-open-source-full-body-tracking-vive-tracker/. [Accessed 10 April 2019].

Uploadvrcom. 2019. UploadVR. [Online]. [2 May 2019]. Available from: https://uploadvr.com/facebook-f8-2019-body-tracking/

Wareablecom. 2019. Wareable. [Online]. [3 May 2019]. Available from: https://www.wareable.com/vr/best-vr-headsets-2017

Uploadvrcom. 2019. UploadVR. [Online]. [16 May 2019]. Available from: https://uploadvr.com/how-vr-tracking-works/

Optitrackcom. 2019. OptiTrack. [Online]. [18 May 2019]. Available from: https://optitrack.com/motion-capture-virtual-reality/

Neuronmocapcom. 2014. Perception Neuron by Noitom. [Online]. [18 May 2019]. Available from: https://neuronmocap.com/products/perception\_neuron

Yostlabscom. 2019. Yost Labs. [Online]. [18 May 2019]. Available from: https://yostlabs.com/priovr/

Tvicoio. 2019. Tvicoio. [Online]. [18 May 2019]. Available from: https://tvico.io/

Vrheadscom. 2016. VRHeads. [Online]. [19 May 2019]. Available from: https://www.vrheads.com/exposing-magic-behind-htc-vive-controller

Unity3dcom. 2019. Unity. [Online]. [19 May 2019]. Available from: https://unity3d.com/unity

Ben lang. 2017. Road to VR. [Online]. [19 May 2019]. Available from: https://www.roadtovr.com/latest-vive-shipping-with-tweaked-base-stations-redesigned-packaging/

Sean buckley. 2019. Gizmodo. [Online]. [19 May 2019]. Available from: https://gizmodo.com/this-is-how-valve-s-amazing-lighthouse-tracking-technol-1705356768

Roguecodecoza. 2019. Roguecodecoza. [Online]. [19 May 2019]. Available from: http://blog.roguecode.co.za/Unity3D-Mesh-Collider-vs-Box-Collider

Unity technologies. 2019. Unity3dcom. [Online]. [19 May 2019]. Available from: https://docs.unity3d.com/ScriptReference/Rigidbody.html

Kirill karev. 2018. Hacker Noon. [Online]. [21 April 2019]. Available from: https://hackernoon.com/behold-the-next-generation-vr-technology-part-2-full-body-tracking-1210ab27ebe4

Medium. 2017. Tonyvt skarredghost. [Online]. [25 April 2019]. Available from: https://medium.com/@SkarredGhost/why-full-body-virtual-reality-and-how-b2cf6783be8c

Arstechnicacom. 2019. Ars Technica. [Online]. [19 May 2019]. Available from: https://arstechnica.com/gaming/2016/09/unity-at-10-for-better-or-worse-game-development-has-never-been-easier/

Dailydotcom. 2018. The Daily Dot. [Online]. [19 May 2019]. Available from: https://www.dailydot.com/debug/unity-deempind-ai/

Unity3dcom. 2019. Unity Technologies Blog. [Online]. [19 May 2019]. Available from: https://blogs.unity3d.com/2014/09/03/documentation-unity-scripting-languages-and-you/

Unrealenginecom. 2019. Unrealenginecom. [Online]. [19 May 2019]. Available from: https://www.unrealengine.com/en-US/features

Sundaysundaeco. 2018. Sunday Sundae. [Online]. [19 May 2019]. Available from: https://sundaysundae.co/unity-vs-unreal/

Educbacom. 2018. EDUCBA. [Online]. [19 May 2019]. Available from: https://www.educba.com/c-plus-plus-vs-c-sharp/

Epicgamescom. 2019. Epic Online Services. [Online]. [19 May 2019]. Available from: https://dev.epicgames.com/en-US/services

Capitalnumberscom. 2017. Capital Numbers. [Online]. [19 May 2019]. Available from: https://www.capitalnumbers.com/blog/which-is-a-better-game-engine-for-beginners-unity-3d-or-unreal/

Unity technologies. 2019. Unity3dcom. [Online]. [19 May 2019]. Available from: https://docs.unity3d.com/ScriptReference/Quaternion.html

Hackadayio. 2019. Hackadayio. [Online]. [19 May 2019]. Available from: https://hackaday.io/projects?tag=kinect

Unity technologies. 2019. Unity3dcom. [Online]. [20 May 2019]. Available from: https://docs.unity3d.com/2017.4/Documentation/Manual/MonoDevelop.html

Unitycom. 2019. Unitycom. [Online]. [20 May 2019]. Available from: https://answers.unity.com/questions/45892/unity-unit-scale.html

Skarredghostcom. 2016. The Ghost Howls. [Online]. [20 May 2019]. Available from: https://skarredghost.com/2016/12/02/the-difference-between-kinect-v2-and-v1/