School of Computing
FACULTY OF ENGINEERING

UNIVERSITY OF LEEDS

Vive Virtual Reality Technology Demonstration

Khen Cruzat and Philip Nilsson

Submitted in accordance with the requirements for the degree of Computer Science

2016/2017

The candidate confirms that the following have been submitted.
<As an example>

Items	Format	Recipient(s) and Date
Final Report (2 copies)	Report	SSO (DD/MM/YY)
Final Report (digital)	Report	VLE (DD/MM/YY)
Project Code	GitHub Repository	Supervisor, Assessor (DD/MM/YY)
User Manual	Report Appendix	Client, Supervisor (DD/MM/YY)

Type of project: Software Product

The candidate confirms that the work submitted is their own and the appropriate credit has been given where reference has been made to the work of others.

I understand that failure to attribute material which is obtained from another source may be considered as plagiarism.

Khen Cruzat

Philip Nilsson

Summary

<Concise statement of the problem you intended to solve and main achievements (no more than one A4 page)>

Acknowledgements

<The page should contain any acknowledgements to those who have assisted with your work. Where you have worked as part of a team, you should, where appropriate, reference to any contribution made by other to the project.>
Note that it is not acceptable to solicit assistance on ‘proof reading’ which is defined as the “the systematic checking and identification of errors in spelling, punctuation, grammar and sentence construction, formatting and layout in the test”;
see http://www.leeds.ac.uk/gat/documents/policy/Proof-reading-policy.pdf.

Contents

Chapter 1

Introduction

1.1 Problem Statement

The goal of the project is to produce a technical demo for the HTC Vive to be used by the university to showcase development skills for virtual reality. This demo would be used in the School of Computing open days.

1.2 Client Background

The client for this project is the School of Computing in the University of Leeds.

1.3 Problem Background

The School of Computing wanted this project to be done as currently they own virtual reality hardware, however they do not have any software made by University of Leeds students to show to potential students. They are currently using software bought online in order to show the capabilities of the virtual reality hardware. They would prefer it if the software that they used is made by students from the School of Computing.

1.4 Project Aim

The aim of this project is to create a technical demo for the School of Computing, using the HTC Vive. This demo should appeal to prospective students, as well as appealing to people in the industry. This means that the project has to be both technical, for the industry, and interesting, for the prospective students.
To make it technical enough for the people in the industry features have been added that are not trivial to implement in Unreal Engine 4.
To make it interesting for the prospective students, the demo has to have good gameplay and an interesting concept behind it.

1.5 Possible Demo Idea

One possible idea for the demo would be to combine the Towers of Hanoi with graph flow. These could be combined by having a generated landscape with a randomly generated graph on it, matching the flow of the terrain. This graph's edges would be rivers, or ditches with water running through them, and the nodes would be either river intersections or pools of water. This would be merged with the Towers of Hanoi by using the Towers of Hanoi system as a dam, to block off flow to a certain river, or by moving the disks you could control the amount of flow. This would work by having the disks stack upside down, with the smallest disk at the bottom. This is done in order to accommodate the shape of the ditch. The less disks that are blocking the river, then the more flow it would have.
The goal of this demo would be to keep all the plants at each node alive. The plants would be considered alive if they got the right amount of water. Too much water they would die and too little water they would die too.

This would be a possible demo idea as it implements several features that are non-trivial in Unreal Engine. Tasks are classified as non-trivial if they cannot be done in engine. It also demonstrates two aspects that are covered in the computer science course, which are the Towers of Hanoi, and Graph Flow.
These features are:

- Running water
- Water Collision
- Having the plants be affected by the amount of water
- Randomly generated river "graphs"
- Towers of Hanoi logic for flow control

The trivial tasks, that are done in-engine, would be:

- Generating terrain and landscapes
- Simple Gesture Controls
- Simple virtual reality gameplay (including teleport mechanic)
- Physics
- Flowers on the terrain

1.6 Deliverables

1. A link to the full code repository on GitHub
2. An instruction manual, detailing how to compile the code, the objectives of the technical demo, and how to control the technical demo
3. Project Report

The reasoning behind these deliverables are:
The code is needed so that the assessors can see what has been for the project, and this will show all the progress that has been made on the software over the course of the project, and how each feature was implemented. This will be on the version control site that is being used for the project, which is GitHub. The Version Control page will be delivered so that the software engineering project management side of the project can be assessed.
An instruction manual was decided on so that the assessors know how to compile the code properly, so that they can test the software, and it will also detail the controls and the objective behind the game.
The project report should be delivered as it provides insight into the inner workings of the project. It also shows the knowledge that the authors gained from doing the project.
These will be the only deliverables as they fully encompass all the work done during the project.

Chapter 2

Background Research

2.1 Virtual Reality

Virtual Reality is a technology that has been around since the early 19th century, although in a primitive form through the use of stereoscopic photos [12]. Stereoscopic photos work by using two photos that are taken of the same place but are slightly offset from each other, as can be seen in 2.1. This creates an illusion of depth for the person viewing the images, when viewed through a stereoscope. A stereoscope is a viewing device that only allows one eye to see one of the two images, so each eye sees a similar, yet different image, and this gives the illusion of depth. Stereoscopic vision is the same technology used in current Virtual Reality headsets although now the images are moving.

Figure 2.1: Example of a stereoscopic image.

Virtual reality platforms have been released aiming to provide an immersive experience to consumers. There are many varieties currently available and they can be simply separated in to the two categories: mobile and desktop. Mobile experiences such as the Google Cardboard and Samsung's Gear VR target the audience which already own a compatible mobile device thus eliminating the cost of hardware found in higher end platforms. Through the use of the phone's built in gyroscope and accelerometer, crude head tracking can be achieved to emulate a virtual world.

High end virtual reality platforms target enthusiasts and early adopters of cutting edge technology due to its premium price and high computer hardware requirements in order to run it. Currently there are two virtual reality headsets that are seen as the devices that give highest immersion and these are Facebook's Oculus Rift and HTC's Vive. These will be discussed later in this chapter.

2.1.1 Mobile VR

On the market right now there are two different Mobile Virtual Reality hardware. There is the Samsung Gear VR and the Google Cardboard.

Google Cardboard

The Google Cardboard is the cheapest Virtual Reality headset out on the market right now, but it does come with the least features out of them. The cardboard viewer is a stereoscope made out of cardboard. It contains two 40mm focal lenses that are designed to give a distortion when looking through them, which is counter-acted by the distortion from the application[4].
To use the Google cardboard you would need to install the cardboard application on your compatible phone and then place your phone inside of the Google Cardboard. Once the phone is inside the Cardboard it uses the phone's inertial measurement unit to track head movement. This does have limitations however as the Google Cardboard does not track displacement if the user was to walk in any direction.
The Google Cardboard still uses the technology of stereoscopic images, as can be seen in 2.2. Although it now does it with moving images, which creates a more immersive experience.

Figure 2.2: Image showing the Cardboard demo application

The Google Cardboard was not chosen to the virtual reality device for this project as there are many drawbacks to it, and as such it does fully demonstrate all the features present in modern technology for virtual reality. The drawbacks to the Google Cardboard are:

- No displacement tracking, making it less immersive than the other options
- Only one input method, a button on the cardboard which acts as a screen press.

Samsung Gear VR

The other mobile Virtual Reality headset on the market is the Samsung Gear VR. The Samsung Gear VR is slightly more expensive than the Google Cardboard, and as expected with the price increase, it

comes with more features compared to the Google Cardboard.
The Samsung Gear VR uses the same technology as the Google Cardboard in the sense that it uses stereoscopic imaging to create the illusion of depth. This is done in the same way for both VR devices, by inserting a compatible phone into the phone holder in the headset, and then showing the stereoscopic images on the phone screen. As seen in 2.3 the Samsung gear VR uses the same stereoscopic technology as the Cardboard uses, as seen in 2.2.
The Samsung Gear VR also uses an inertial measurement unit to detect head movement, similar to the Google Cardboard. The Samsung Gear VR uses an inertial measurement unit contained in the headset, rather than using the attached phone's inertial measurement unit. The inertial measurement unit contained in the headset is more accurate, has lower latency, and is better calibrated than standard phone inertial measurement units, as it uses the same I.M.U. as the Oculus Rift. This I.M.U. is more accurate as it has a higher sample rate than internal phone I.M.U.s and therefore gives it more values to use, so that it can more accurately detect erroneous values.

Figure 2.3: Image showing the Samsung Gear VR menu

The Samsung Gear VR has a few extra features compared to the Google Cardboard, for example when a phone is placed inside the Galaxy Gear VR it needs to be connected by a micro-usb connection, which allows the headset to have more input methods to the phone, as well as giving access to the headset's I.M.U. The extra input methods that the Gear VR has access to are:

- A home button, which works the same as the home button on Android phones.
- A back button, which works the same as the back button on Android phones.
- A touch pad, which works by swiping to move across menus, and tapping clicks the highlighted item in a menu.

Figure 2.4: Image showing the hardware controls on the Samsung Gear VR

The Samsung Gear VR will not be used for this project as again it has several drawbacks, which are:

- It only tracks rotational movement, not displacement, which makes it less immersive than the other options

- The Samsung Gear VR has very primitive control, which are only the buttons and touchpad on the side of the headset

2.1.2 Oculus Rift

The Oculus Rift was the first of the two to be released and is inferior in terms of the level of immersion that can be achieved, as currently Oculus only supports interfacing with the virtual world through a third party traditional controller that simply uses buttons and joysticks. The Oculus Rift tracks by using the single camera to pick up infrared light that is emitted by points on the headset. These can be used to track the headset as they blink in a specific pattern, which the sensor knows, and it then uses that to determine the position the headset is in.

2.1.3 HTC Vive

The HTC Vive works using two base-stations. These emit lasers in an alternating pattern, between vertical and horizontal. If these lasers hit a sensor on the headset or controllers, they emit a pulse. By tracking the timings of the laser sweeps and the emitted pulses, the tracking system can use trigonometry to find the position of the location of every sensor on the devices [11]. The HTC Vive has two settings, either a sitting or a standing mode. in the sitting mode, it works similar to the oculus Rift, in that a console controller is used to control the game, where as in the standing mode, it uses it's own controllers, which are tracked by the base stations and provide a more immersive experience as you can interact with objects in the game by using these controllers to pick things up by moving your hands to where the object is in game.

2.2 Development Environments

To develop on the HTC Vive there are two options which are currently supported and these are the Unity engine and the Unreal 4 engine. These game engines has native support for SteamVR which is the platform developed by Valve that powers the Vive. Both are free to be installed and just requires a simple registration to their respective websites.
Unity supports the C# programming language for development whereas Unreal Engine uses C++ on its current version of the software. Another aspect where it differs is Unity is mainly used to develop games for mobile devices such 2D platformer games. Unreal Engine is mainly used for desktop, AAA games which makes it more suitable for virtual reality with games running on Unreal generally looking better. Unreal Engine includes many features built in such as particle effects simulation, terrain, lighting and shading [1]. With SteamVR being natively supported, simple virtual reality features such as gesture recognition and teleportation movement mechanic can be easily implemented in to a game.
For this project Unreal Engine will be used as the development environment. This is because the group is more comfortable with using C++ so development will be quicker with Unreal rather than trying to use a language with little familiarity.

2.2.1 Unreal Engine 4

The Unreal Engine offers many features for people with little to no programming experience, these will not be used however, as they do not demonstrate any Computer Science expertise, one of the many features that will not be used is the drag and drop interface that is packaged in Unreal Engine to develop simple software quickly, along with the basic templates for various types of games. These are templates for popular genres, for example First Person Shooter and Sidescroller. For this project the group will implement their own features and backend for the genre that is picked, in order to demonstrate their ability to code for the HTC Vive. A select list of the features that Unreal Engine implements already are as follows:

Unreal Engine 4 Features

- Particle Effects Simulation (Visual Effects)
- Procedural Foliage
- Landscaping/Terrain
- Lighting
  - Directional
  - Point
  - Spot
  - Sky
  - Shadow Casting
- Shading
- Post Process Effects
  - Bloom
  - Ambient Occlusion

  - Colour Grading
  - Depth of Field
  - Lens Flares
- Material Effects
- Fog Effects
- View Distance Culling
- Distance Dependent Level of Detail Models
- Physics Simulation
- Level Streaming (Ability load and unload map files into memory and toggle their visibility)
- Basic Templates
  - First-Person
  - Third Person
  - Side Scroller
  - Vehicle
- Artificial Intelligence System
- Audio System
- DirectX 11 & 12 Features
  - Full-scene HDR reflections
  - Per Scene Dynamic Lights
  - Physically Based Shading

Unreal Engine's full list of features can be found on their documentation site [2]

Chapter 3

Requirements

3.1 Client Requirements

The client has asked us to produce this technology demonstration in order to have a piece of software for the HTC Vive to demonstrate to students and has been developed by a University of Leeds student. The requirements that were given by the client were to appeal to the target audiences (who will be discussed in the next subsection), and to fully utilise the functionality of the HTC Vive in order to properly demonstrate its capabilities.

3.1.1 Target Audience

The demo would be targeted towards potential students looking to apply to the University of Leeds, as it will be shown to the students on Open days in order to gain interest from them. It would also have to appeal to people and companies in the gaming industry. As the more interest the School of Computing can gain from them, the more potential projects they may have for the school. With these target audiences in mind the demo should be technical to impress the gaming industry, while balancing it with being interesting for the potential students.

3.2 Feasibility Assessment

3.2.1 Feasibility

To support the development of the project, a reserved space where the HTC Vive can be permanently set up for the duration of the project is required. This reserved space would ideally be a room that meets the space requirements stated above for room-scale experiences, so that all the capabilities of the Virtual Reality hardware can be used. A computer which meets the hardware requirements is also needed in order to run the HTC Vive software, This computer must be running Windows since HTC Vive currently only supports this operating system [5]. Also, a copy of Unreal Engine 4 game engine must be installed which is free to be downloaded. Unreal Engine 4 is chosen over Unity since the members of the group are more familiar with developing in C++ which Unreal Engine supports rather than C# which Unity supports, although both of these game engines provide native support for virtual reality developments.

A possible solution for meeting the hardware requirements is to use a personal machine. A laptop is available with the specifications below which just meets the requirements for the graphical power. Development can be done on our own personal Windows machines and can be tested with the Vive using the laptop in the reserved room. The Vive is not required for conducting simple tests, but it is needed for identifying issues such as scaling and user input with motion controllers.

3.2.2 Technical Specifications

Hardware Requirements

- Graphics card: NVIDIA GeForce GTX 970 /Radeon R9 280 equivalent or greater
- Processor: Intel Core i5-4590 equivalent or greater

- RAM: 4GB+
- Video Ports: HDMI 1.4, DisplayPort 1.2 or newer
- 1x USB 2.0 port
- Room-Scale Space: 2 meters by 1.5 meters

These requirements can be found on the official HTC Vive page on the Steam Store [10]

Chapter 4

Project Management

4.1 Methodology

The methodology that will be used for this project is the Agile methodlogy. This methodlogy will be used for this project as following the agile methodlogy gives you a working product earlier in the building phase. Agile also requires a weekly standup to be held, this will keep the project on schedule, as during this standup meeting the week's progress on the project will be checked and evaluated.

4.2 Schedule

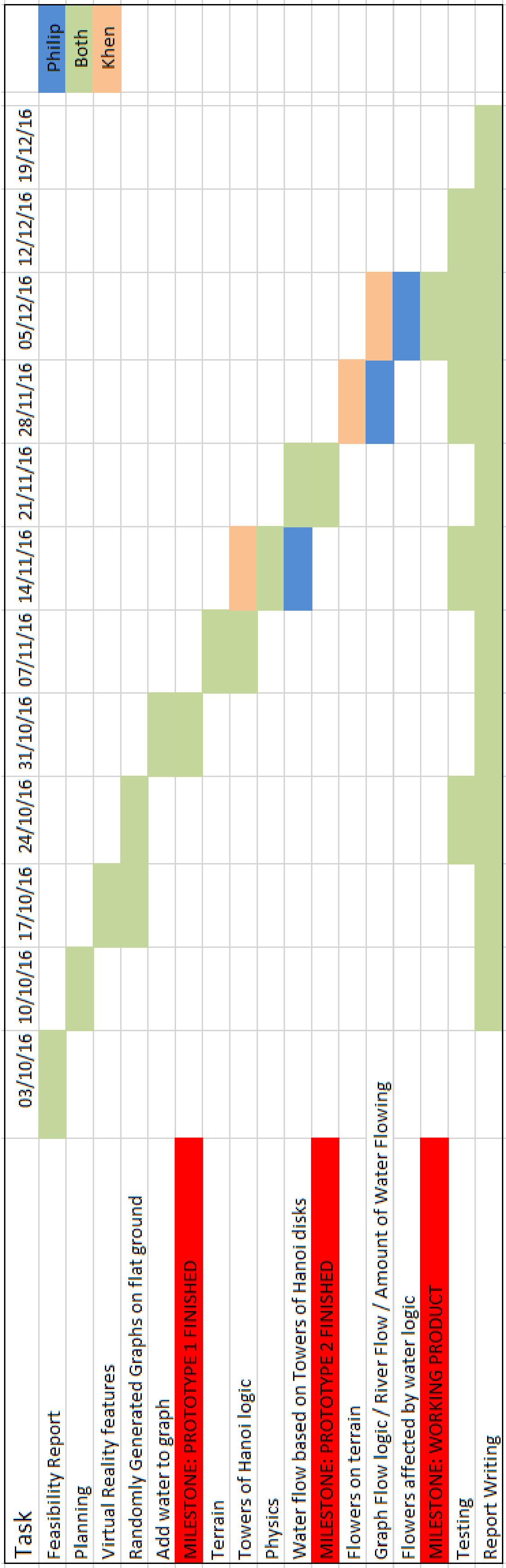

Figure 4.1: Gantt Chart showing the work that will be done each week.

Writing Milestone	Date For
Layout Done	14/11/2016
Chapter 1 - Introduction	21/11/2016
Chapter 2 - Background	28/11/2016
Chapter 3 - Requirements	28/11/2016
Chapter 4 - Project Management	05/12/2016
Chapter 5 - Planning and Design	05/12/2016
Chapter 6 - Implementation	12/12/2016
Chapter 7 - Testing and Evaluation	12/12/2016
Chapter 8 - Conclusion	12/12/2016
First Draft Complete	12/12/2016
Report Finished	23/12/2016

4.3 Version Control

The Version control for this project will be done on github. Version control software will be used for the project in order to accurately document any and all changes that will be made to the code.

4.4 Risk Assessment

A risk involved with this project is the availability of the HTC Vive headset as well as its peripherals. Since the tech demo will be making use of the Vive as the virtual reality hardware, it is essential that a Vive is at hand to be used for testing the demo. The school currently owns 2 Vive packages and since the school is currently only using them for demoing purposes of off the shelf demos and not for development, a Vive should be available for the development in this project. If there is no access to a HTC Vive headset, a contingency plan has been made, which is outlined in 4.4.1.
Another risk that may occur is that the Vive has certain requirements for the space it needs to use for a room-scale experience. The Vive takes some time to set up since it has to recalibrate to the environment every time it is set up again. This means a dedicated room where the ViveâĂŹs base stations can be left set up is ideal. If a big enough room is not available, Vive has an option to use a standing/seated experience instead. This will means that user movement cannot be tested and movement in the virtual world will have to be restricted to ViveâĂŹs teleportation mechanic. The user looking around the world will still be supported with this version.
One other factor that needs to be taken in to consideration is the availability of computer hardware that is powerful enough to run and develop on the HTC Vive. Since the HTC Vive has high minimum requirements to run it, finding a computer that is sufficient and can be used when needed can become an issue.

4.4.1 Contingency Plan

In the case that there is no access to a Vive Headset for this project, due to the risks mentioned before, a contingency plan has been made. The plan would be to make a desktop application using the same premise, rather than being able to use the Vive for the game. This desktop application would have the same features as the Vive version, although without the VR features, i.e. looking around using the headset and interacting with objects using the controllers.
The plan for dealing with if there is no room available is that a seated experience could be developed without need a room set-up. This would just involve setting up one camera above the computer and it would only track head movement. A standard game controller would have to be used for this set-up.

For the risk that there is no high-end computer available, a high-end laptop that can run the software has already been acquired and will be used if there is no other computer available.

Chapter 5

Planning and Design

5.1 Game Design Process

The game design process started by using the HTC Vive to research what kind of games were already on the market for the HTC Vive. The games that were SteamVR Demo[9], The Lab[8], Space Pirate Trainer[6], and Elite Dangerous[3]. This research was done in order to give a feel for the HTC Vive and it's capabilities, this would give a clearer understanding to which kind of games are more suited to the Virtual Reality platform.
The next step in the game design process was to brainstorm ideas of what kind of game would be made for the project. The ideas would be split into four categories, these four categories are the theme of the game, the features of the game, the genre of the game, and the benefits of the game.
The brainstorm was then categorised in a separate table, combining the ideas that have relevance together, this table can be seen in **??**. Then using the updated table of ideas, the ideas were narrowed down until one that could feasibly done was chosen out of all the ideas on the white-board. This idea was to combine the Towers of Hanoi and Graph Flow in a puzzle game. These two ideas were combined in order to make a technology demonstration that is very specific to the School of Computing in the University of Leeds, as these are both topics of study during the Computer Science course in the university of Leeds.
A goal for the game then had to be designed. The goal of the game was decided to be having the right amount of flow run down to set nodes, where flowers would be. If these flowers have too much water flow they would look dead, and if they had too little water flow they would also look dead. They would only look alive if the water flow was the correct amount. The game is won when all the flowers are "alive" at the same time.
This idea would implement many non-trivial features in the Unreal Engine, such as:

- Randomly generating a graph
- Generating a terrain based on the graph
- Towers of Hanoi logic for flow control
- Having plants being affected by the water flow
- Changing the appearance of the water, based on the flow of that edge of the graph

5.2 Virtual Reality Features

The Virtual reality Features that will be used are:

- Using the Head Mounted Display in order to look around the virtual world.
- Using the controllers to pick up the Towers of Hanoi disks, and place them down.
- Using the controllers to select a teleport location to move.

5.3 Graph Flow

Graph Flow will be used in this technical demonstration as the main objective of the game. The objective of the game will be to get the flow to match the goal flow. This will be done by blocking off certain rivers to redirect flow from one river into another. The goal flow will be calculated in development, and each level that is created will also have a goal state completed.

5.4 Towers of Hanoi

Towers of Hanoi will be used in the demo to block off the graph flow from nodes to other nodes. This will be done by having rods in the river at each split in the graph. These will be placed at the output of each split in the river. The Towers of Hanoi disks will then be placed on top of these rods if the flow needs to be blocked. The Towers of Hanoi disks will only be allowed to be placed in the opposite order compared to the way the normally are. This would mean that a larger disk can only be placed on top of a smaller disk. This order had to be inverted to the standard as rivers are narrower at the bottom, compared to the top of the river ditch.

5.5 Plant Survival

The Plant Survival will be used as a measure for the player to determine how close he is to solving the puzzle. This will be the indicator to the player if the flow at that node matches the pre-determined flow for the level

Chapter 6

Implementation

6.1 Development Environment

The development of this project was done in various environments, as there was many different aspects to this project.

6.1.1 Vive Hardware

For this project, the Vive Hardware was set up in a dedicated room (The Virtual Reality Lab in the School of Computing). This allowed development to be done on the HTC Vive without having to set up the sensors and calibrate the hardware everytime that development had to be done, which maximised the amount of development time that was available.

Figure 6.1: Room used for testing the application

This room was chosen as it was unused and met the space requirements for the HTC Vive.

6.1.2 Game Engine

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

[7]

6.1.3 Visual Studio

6.1.4 Windows

6.1.5 Out of Engine Development

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

[7]

6.2 Random Generation of Graphs, Rivers and Terrain

6.2.1 Graph Generation Original Method

Point Insertion

The original idea to do graph generation was to start with a square, using each corner as a node. These nodes would then be connected as shown in 6.2.

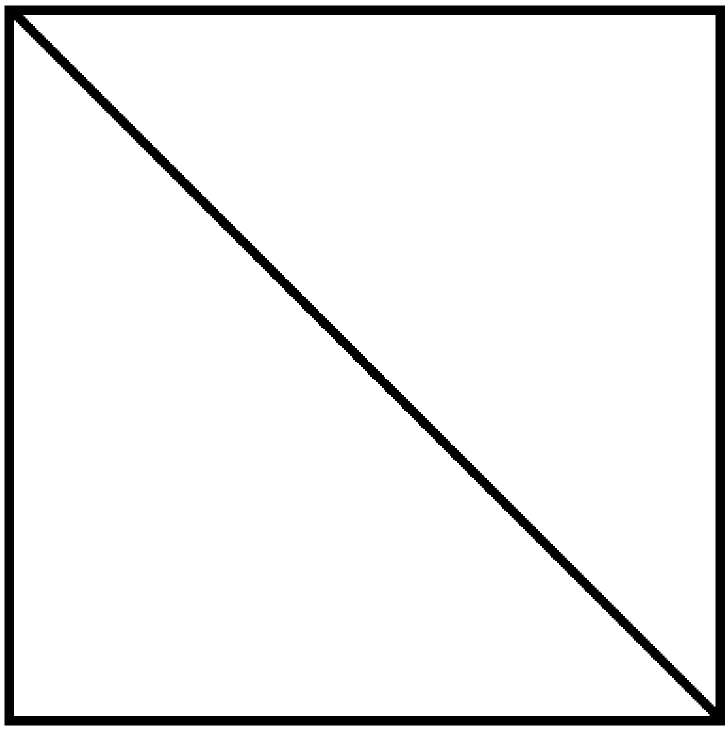

Figure 6.2: Original Connections

Points would then be inserted into this square, using a random x and y value. The triangle that this node is in would then be found using the cross product. This was done by checking the cross product of the point and the triangle, for each triangle that is in the graph. The point would then be connected to the corners of this triangle.

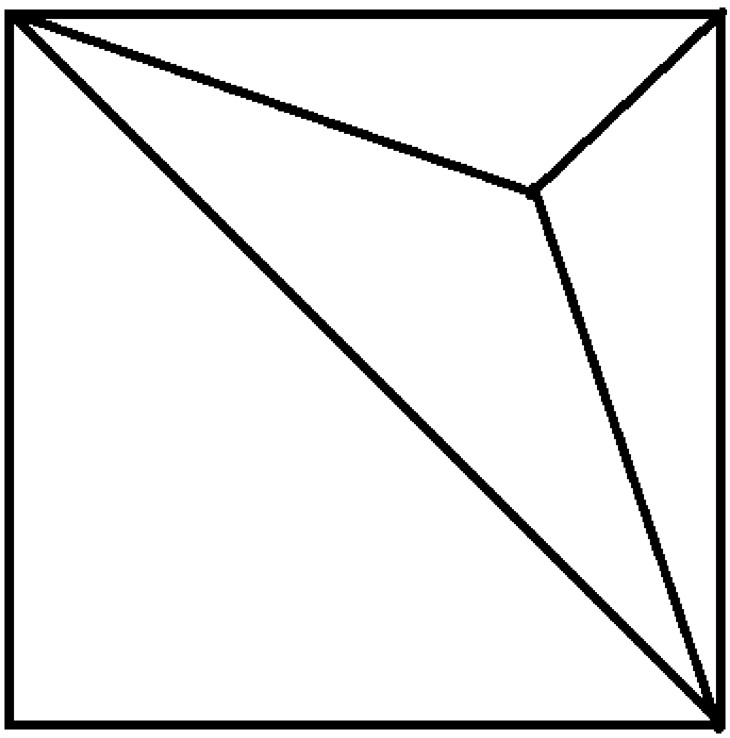

Figure 6.3: Example of running the point insertion algorithm for one iteration.

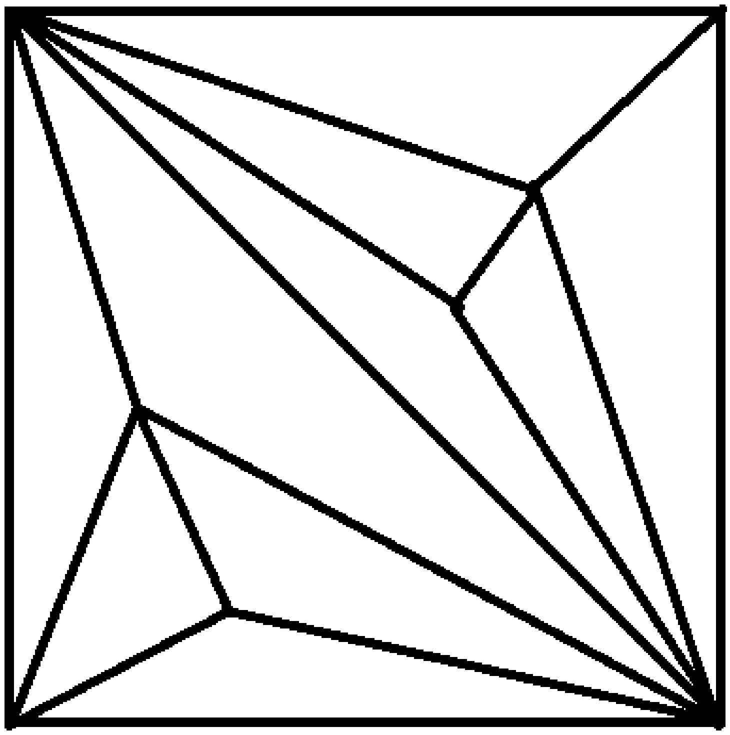

Figure 6.4: Example of running the point insertion algorithm for four iterations

This approach guaranteed a connected graph to begin with. This approach did not work however as if a node on the bottom-left side of the graph needed to be connected to a node on the top-right side of the graph, the the connection would have to be made through either the top-left node or the bottom-right node, as there was no other way to pass through to the other side of the graph. The approach was then modified slightly to start with an extra node node in the middle of the square, allowing another way to pass through the other side of the graph, this is seen in 6.5.

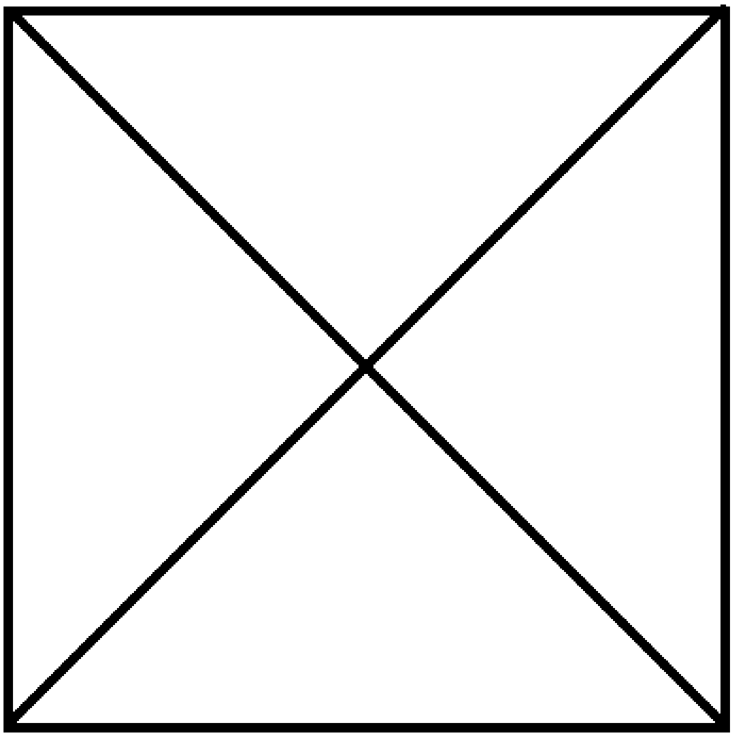

Figure 6.5: Image showing the second method's starting connections

This approach also did not work, as the addition of the extra node did not provide enough of relief for the connections between the two sides of the graph, and the connections would occasionally still go through the top-left or bottom-right node. The next approach was adding seveeral nodes along the diagonal and connecting them to the corners, similarly to how the middle node was added. The third approach can be seen in 6.6.

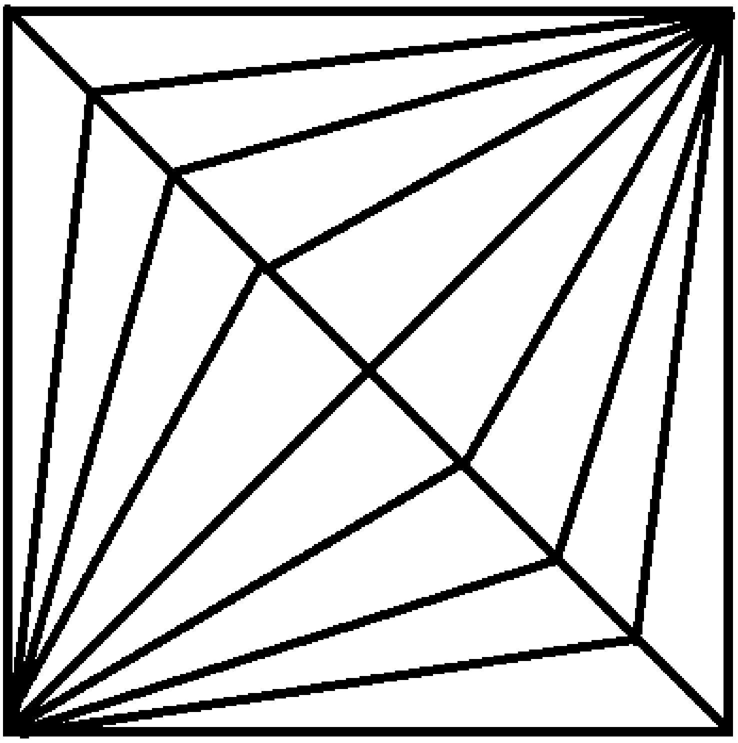

Figure 6.6: Image showing the third method's starting connections

Connections and Weight Matrix

The connections were found by looping through the array that stored the triangles and finding the edges of the triangles. These edges were stored in a connection matrix, to be used when making the weight matrix.
The weight matrix was made using manhattan distance between the two connected points as the heuristic. The manhattan distance is the x distance between the two points plus the distance in the y direction. The manhattan distance was calculated using the following formula:

$$Manhattan = (point1.x - point2.x) + (point1.y - point2.y)$$

The manhattan distance then has to be checked to see whether it is positive or not. If it is positive the result is put in element of the matrix representing the first point to the second point, if the result is negative the result is instead placed in the element representing the second point to the first point. This is repeated for each connection in the connection matrix produced before.

Generating Rivers

The river generation starts off with picking the start and end nodes for the rivers, this is done by randomly selecting a set amount of nodes, determined by a variable set in code.

6.2.2 Terrain Generation

6.3 User Interactive Reverse Towers Of Hanoi

The logic for the reverse Towers of Hanoi was implemented by using collision boxes on the rod actor to act as trigger when Towers of Hanoi discs are placed and removed. This method and actor was used as the basis and control for the Towers of Hanoi logic. To differentiate between the discs, three different actors were created for each size of the Towers of Hanoi discs: small, medium and large.

A collision box covers the whole rod with a size that is bigger thane the size of the hole in each disc is set to BlockAll so a disc cannot be placed through the rod. When an actor comes in contact with the collision box, this triggers the rod actor's OnHit collision event. Within this event, it identifies the size of the actor through the name of the actor's class and does a check with the array of discs in which that specific rod currently contains. This simple check will only allow discs which are larger to be placed on top of smaller discs and also makes sure that the array doesn't add disc actors that are already in the array. When the check is passed, the collision box's collision profile name is set to OverlapAll to allow the disc that triggered the event to be placed in that rod otherwise it sets it back to BlockAll. Furthermore, it adds an upward force to the disc to show that the action the player is trying to complete is invalid. This force is only added when the player is not holding the disc to avoid issues when the player is holding the disc and are moving/looking around and accidentally touching a rod with the disc.

A second collision box that is slightly larger than the first was used to handle the adding and removing of discs to the array which contains all the disc actors which the rod currently holds. When a disc overlaps this collision box it adds it to the array through the OnOverlapBegin event trigger then removes it from the array with the OnOverlapEnd event trigger. When a disc is added to the array, it sets all the discs below it to not allow the player to grab it. This is done to keep with the Towers of Hanoi logic of only the disc at the top of stack being movable.

A bonus feature to help with placing the discs was implemented with the 'snap to rod' feature. When a disc is allowed to be placed on the rod so the disc is larger than the disc below it, then as soon as the OnOverlapBegin collision event is triggered, the disc is teleported to the top of the rod with its rotation reset so it can slide down the rod. This is useful as it can sometimes be bothersome to the player to place the disc exactly so the rod can fit exactly through the hole of the disc. This feature is only in effect when the disc is not currently being held by the player to avoid the disc teleporting whilst the player is still holding the disc which causes the disc to still be attached to the player's hand even though the disc is not in range of the hand's grab sphere.

6.4 River Graph Flow

Graph flow logic was implemented by using information generated with the terrain such as the nodes, IDs of rivers and knowing which rivers are connected to which. An actor for each river is created with a mesh using the UProceduralMeshComponent and the vertices supplied in the rivers.txt file. Each river is given an ID with the first part of the 2 numbers of the ID as the node the river is connected to and the last two numbers as the other node the other end of the river is connected to. This means that a series of river connections have the last two numbers of a river ID identical to the first two numbers of the next river in the connection.

The source of the river in the graph has a flow value hard coded and then iterates through its river connection changing the flow value of each river then when it reaches the final river it means that the river has reached a node where it splits in to two. This split has a rod for each river in order for the player to change the flow of each river. With rods with no discs, the flow value is just split simply in half between the two rivers which then follows the same procedure as before and changes the flow value of the rivers in its river connections.

With the case of rods with discs in them, the flow is then affected depending on which discs are used. A small disc will reduce the flow value of the river by a $\frac{1}{6}$, medium disc by $\frac{1}{3}$ and a large disc by $\frac{1}{2}$. This means using all three discs would reduce the flow to 0 so completely blocking it off. Adding or removing a disc from a rod would change the flow value for the river the rod is connected to which then changes the flow value of all the rivers that this river affects.

In order for the player to easily distinguish a difference in the flow of a river when changing them with the discs, the opacity of the material used on the river will change depending on the flow value as a percentage of the original flow value at the source.

6.5 Flow Dependant Flora

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.
[7]

Chapter 7

Testing and Evaluation

7.1 Testing Against Requirements

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

[7]

7.2 Client Evaluation

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

[7]

7.3 Result Evalutaion

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

[7]

7.4 Project Evalutaion

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.
[7]

Chapter 8

Conclusion

8.1 Conclusion

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.
[7]

8.2 Future Work

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.
[7]

8.3 Personal Reflection

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.
[7]

References

[1] Epic Games. About unreal engine 4.

[2] Epic Games. Engine features.

[3] Frontier Developments. Elite dangerous.

[4] Google. Google vr sdk for unity.

[5] C. Hoffman. The htc vive and valve's steamvr don't yet support linux and steamos.

[6] I-Illusions. Space pirate trainer.

[7] D. Parikh, N. Ahmed, and S. Stearns. An adaptive lattice algorithm for recursive filters. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 28(1):110–111, 1980.

[8] Valve. The lab.

[9] Valve. Steamvr.

[10] Valve Corp. Htc vive.

[11] Valve Corp. Steamvr tracking.

[12] C. Wheatstone. Contributions to the physiology of vision. part the frist. on some remarkable, and hitherto unobserved, phenonmena of binocolur vision. *Philosophical Transactions*, 128:371–394, 1838.

Appendices

Appendix A

External Material

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

36

Appendix B

Ethical Issues Addressed