

University of St. Andrews
School of Computer Science

Tower Defence Game In Virtual Reality

Yves Fricker

Supervised by: Professor Ian Miguel

Submitted in part fulfilment of the requirements for the degree of
Bachelors of Computer Science of the University of St. Andrews, December 2017

Abstract

A tower defence game is, at its core, a game where the player has to stop a horde of enemies from travelling from one point of the map to another. To do so the player can erect different types of towers or turrets which shoot projectiles that destroy the enemy. This project focuses on executing a tower defence game in virtual reality. The key feature of which is that a player can micromanage the towers firing at the enemy and take part in the combat themselves through a first person perspective. The report initially highlights the core concepts that define a tower defence game and exposes the reader to the history of the genre. Thereafter, this report documents the software engineering process, game design and game implementation. To finish, a critical evaluation will be detailed in respect to the project's initial specification as well as the feedback acquired through user testing.

Declaration

I declare that, unless indicated by appropriate referencing, all the work detailed in this report is my own. Furthermore, the sole assistance I received on this project was that of my supervisor Dr. Ian Miguel. No part of this project has been previously submitted for assessment at this university or elsewhere. The text is 8718 words long excluding references and any appendices. By submitting this project to the University of St. Andrews I give permission for it to be made available for use in accordance with the regulations of the University Library. I also give permission for the title and abstract to be published and for copies of the report to be made and supplied at cost to any bona fide library or research worker, and to be made available on the World Wide Web. I retain the copyright in this work.

Yves Fricker:



Senior honours student of the St. Andrews School of Computer Science

Contents

Abstract	i
1 Introduction	1
1.1 The Tower Defence Genre	1
1.2 Virtual Reality	2
1.3 Game Overview	5
1.4 Project Baseline	6
1.5 Ethical Considerations	7
2 Software Engineering Process	8
2.1 General Approach	8
2.2 Unity as a Game Engine	9
3 Game Design	11
3.1 Game Mechanics	11
3.1.1 Creeps	11
3.1.2 Bow and Arrow	14

3.1.3	Towers	14
3.2	Top Level Software Design	15
3.3	Player	16
3.4	HUD	17
3.5	Graphics	17
4	Game Implementation	19
4.1	Path Generation	19
4.1.1	Development of the Algorithm in 2 Dimensions	19
4.1.2	Utilising the Algorithm inside Unity	25
4.2	Creeps	27
4.2.1	Path Finding	27
4.2.2	Spawners	28
4.2.3	Animation	30
4.2.4	Shooting at the Player	31
5	Evaluation	32
5.1	User Testing	32
5.1.1	Spawning locations are obscure	33
5.1.2	Long arrow shots are hard to track and confirm	34
5.1.3	Menu button mapping is not intuitive	34
5.1.4	Towers are not easily recognisable	35

5.1.5	Enemies in the first wave have too much health	36
5.1.6	Gold reward should be displayed on death of an enemy	36
5.1.7	In game tutorial is preferred over game manual	37
5.2	DOER Requirement Comparison	37
5.2.1	Primary Objectives	37
5.2.2	Secondary Objectives	38
5.2.3	Tertiary Objectives	39
5.3	Future Work	40
6	Concluding Statements	41
	Bibliography	42

Chapter 1

Introduction

The goal of this project was to develop a tower defence game that a person can play in virtual reality. This chapter will familiarise the reader with both the tower defence genre and current virtual reality consumer technology. Thereafter, it will give an overview of the final version of the game, followed by the project baseline and ethical considerations that had to be made.

1.1 The Tower Defence Genre

A tower defence game is, at its core, a strategy game that focuses on preventing a group of enemies called *creeps* from travelling from one end of the game map to the other. Traditionally, A player can slow down and destroy his enemies by building towers in predefined locations. These towers will automatically take aim and shoot at individual enemies as they pass within the range of a tower. Often there are different types of towers such as a frost mage tower that can freeze and slow down the enemies as they pass or a cannoneer tower that hurls explosives at the oncoming enemies and inflicts damage to multiple enemies at the same time. There are usually also several types of enemies each with their own weaknesses. A player must therefore consider carefully which towers they place to effectively destroy all types of enemy units. Enemy groups are usually spawned in waves giving the player respite in between destroying waves to place more towers or upgrade existing ones.

Some argue that the genre dates back as far as 1990 with the release of the video game *Rampart* which involved protecting a castle from the projectiles of battleships in a nearby harbour [1]. The player had to build cannon towers to shoot at the ships and defend the castle from harm. While this game may have been one of the ancestors in the evolution of the genre, it is no longer what people typically think of as tower defence as it does not entail enemies travelling along a path from a start position to some destination across the game map. The proliferation of this game type came in two waves. The first wave coincided with the dawn of adobe flash browser games. Tower defence games are not very computational intensive and lend themselves well to the web browser. The first flash tower defence game to gain wide spread success was *Flash Element Tower Defense* with nearly 500,000 users playing the game within one hour of its release [2]. The game implemented many of the genre defining features such as having enemies with weaknesses to different types of towers, requiring the player to think strategically as well as being able to upgrade towers giving the game a sense of progression. The second wave of popularity surged with advent of mobile gaming. Titles such as *Kingdom Rush* (Figure 1.1), *Plants vs.Zombies*, *Field Runners* and *Bloons TD* have seen millions of players toy with the genre. Tower defence games have spread over a wide array of media due to their low computational demands. It is therefore a natural progression to execute this game type in virtual reality which is in of itself very computationally intense.

1.2 Virtual Reality

With a basic understanding of how modern tower defence games work we must now consider what constitutes current virtual reality (VR) consumer technology. In recent years there has been a massive resurgence of the virtual reality industry since its decline in the 90s with the commercial failure of the Nintendo Virtual Boy [3]. Todays consumer VR is spearheaded by 3 large cooperations; HTC, Sony and Oculus. All three companies have developed headsets that contain stereoscopic displays. These displays show each eye a different angle of the virtual scene, thus creating an illusion of three dimensionality. However the way the headsets really differ from one another is by how the player gets to interact with the virtual environment. The



Figure 1.1: Popular TD game "Kingdom Rush" with different types of towers and enemies.

Oculus Rift by Oculus provides a standard Xbox controller for the player to navigate himself or a separate character through a virtual world. This has been critiqued as immersion breaking by the tech community [4] because it creates a disconnect between the movement of your virtual character and your own body. In some cases this can even cause motion sickness. Sony's *PSVR* headset includes a pair of LED tracked joysticks that allows the player to interact with the virtual world at a very basic level much like the *Nintendo Wii* permitted. This leaves us with HTC's product - the *HTC Vive*. The *HTC vive* has been critically acclaimed for its tracking solution [4]. It provides the user with both a tracked headset and two tracked controllers, one for each hand (See Figure 1.2). These tracked devices can be precisely located by the computer anywhere within a predefined perimeter. The tracking perimeter itself is defined by the player depending on where they place the tracking stations. However the minimum tracked perimeter size must be at least be two by two meters and at most five by five meters for the tracking technology to work correctly. Games that operate within this space restriction have been coined *room-scale* VR games [5].

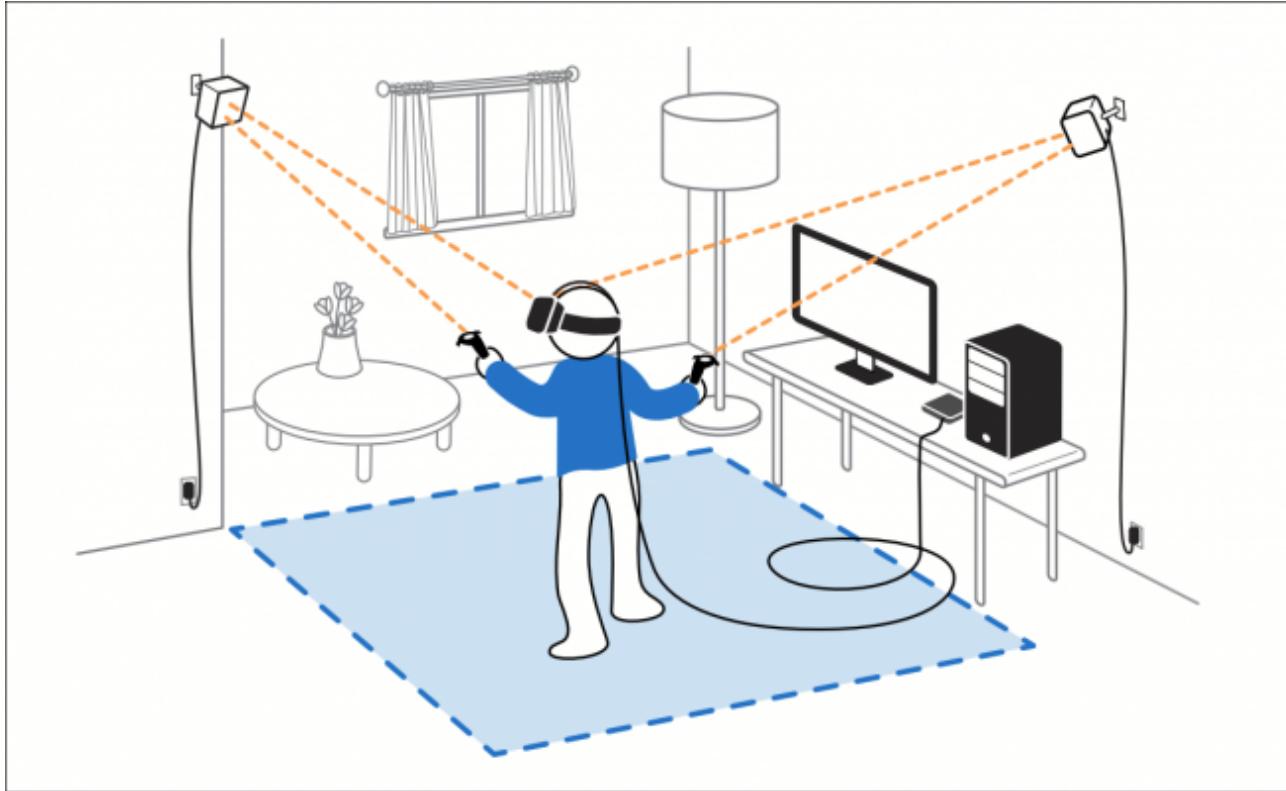


Figure 1.2: HTC Vive tracks both controllers and headset in 3D space

The virtual reality tower defence game detailed in this report was developed for the HTC Vive with the intent of utilising the full potential of room scale VR. The key motivator for developing a tower defence game for the HTC Vive is that in reality the top of a tower could reasonably be around 5 meters in perimeter thus supporting the players virtual immersion. This benefit to immersion may not be apparent until one considers that in most open world VR games a player is prevented from walking further than 5 meters in any direction because of the physical limitations of their room and the capability of the tracking device. The HTC Vive headset communicates this limitation by overlaying a virtual semi-transparent grid on top of the game (Figure 1.3) so that the player does not walk into furniture or real walls. This limitation is immersion shattering because the brain expects to be able to walk further than 5 meters. Unless, of course, the player is placed on top of a very high tower where the virtual area limitation feels very real. This is because the brain does not expect to be able to walk off of a high tower.



Figure 1.3: HTC Vive chaperone system that prevents player from walking into objects.

1.3 Game Overview

The finished game revolves around preventing a set of monsters (Figure: 3.2), called *creeps*, from travelling across the game map along a procedurally generated path (Figure: 4.5). The creeps try to travel from their spawning locations to the holy temple in order to pillage and destroy the treasures inside. The player starts off with a limited number of lives and for every creep that trespasses the temple gates, a life is subtracted from the player. When a player has lost all their lives the game is over. If the player kills a creep they will be rewarded with a preassigned gold value that is proportionate to how difficult it was to kill the creep. Creeps are spawned in waves. For each wave there may be just one spawning point or several spawning points conjuring forth creeps from many corners of the map. Some creeps shoot back at the player resulting in the loss of one life upon an enemy projectile colliding with the player.

The player commands the game from a series of towers that have been placed strategically by the level designer around the map. A player can teleport from one tower to the next using a laser pointer. The player can then proceed to use a bow and arrow to shoot at the creeps in an effort to kill them. The player may use the gold currency earned by killing creeps to summon

allying spirits above each tower in the game. These spirits aid the player by shooting elemental projectiles at the creeps that may apply a debuff effect such as poison. There are currently 4 spirit types in the game; fire, ice, poison and the explosive. The fire spirit is the cheapest and causes damage points to one enemy. The ice spirit projectiles will slow down the enemy by 50% upon impact. The poison spirit will reduce the creeps health by a percentage value at every unit of time. The explosive spirit is the most expensive to spawn and will deal an area of effect damage that deals heavy damage to every creep in a radius of 12 meters.

Additionally the player may spend their coin on upgrading their arrows to turn them into elemental arrows. The player may upgrade their arrows based on whatever elemental spirit is above them by simply reaching into the sky with their arrow hand controller. However, there is a cost of upgrading an arrow that is proportionate to the cost of the spirit. Thereby, fire arrows are the cheapest and explosive arrows the most expensive. Elemental arrows deal the base damage value of a regular arrow on impact with a creep in addition to whatever elemental effect the corresponding spirit would have applied to the creep. Explosive arrows explode on impact with any surface so they are particularly useful as even if the player shoots with poor aim there is a good chance the arrow will explode against a surface nearby the creep and the area of effect damage will harm the monster.

1.4 Project Baseline

A game and physics engine called *Unity* [6] was used to allow for most of the development time to be invested in high level game design and VR features. Unity does not provide any logic for the tower defence mechanisms, it only provides physics simulations and tools to make a 3D game. A framework called steamVR was used alongside Unity as it enables basic features such as being able to assign functions to a key press on a Vive controller. Furthermore, all the graphical assets such as 3D models and animations were derived from the Unity asset store. Game levels themselves were designed and assembled from the individual graphical assets by myself.

1.5 Ethical Considerations

The ethical considerations to make with this project are the same as with any other virtual reality video game. A person who is struggling to cope with reality may be tempted to spend unhealthy amounts of time immersing themselves in virtual reality video games, such as this one. Furthermore, because the HTC Vive implements motion tracking, there exists a risk of users hurting themselves by hitting real life objects that are not present in the virtual world.

Chapter 2

Software Engineering Process

2.1 General Approach

This project was built iteratively in incremental steps. I met with my supervisor once a week to discuss the progress and design decisions I had made throughout the previous week. Even though some requirements were defined at the beginning of the development, strict adherence was never prioritised and a very flexible approach was adopted. This is because the ultimate requirement for almost any game is to be entertaining and it's hard to predict ahead of time whether or not a design decision will be entertaining. Therefore, as part of my approach, many small prototypes were created and play tested by myself to see if a particular design decision was worth pursuing.

As one can imagine building a virtual reality game from scratch without the use of any framework or APIs is not within the time scope of this project. My goal was to create an immersive VR experience and concentrate less on reimplementing the underlying technology and more on the final user experience. To that end I utilised the game engine *Unity* [6] and relied on the steamVR framework which provides an API that interfaces with the HTC Vive hardware.

2.2 Unity as a Game Engine

Unity is a game engine founded in 2005 and has been a popular game engine for independent game developers that want to produce graphically demanding games [7]. It was the first public game engine to support the HTC Vive. The decision to work with Unity had a drastic impact on my software engineering process. Most of my programming experience has previously taken place under the inheritance paradigm that is so common in standard object orientated program. While Unity does support object orientated programming, it does not encourage creating subclasses of other classes. Instead, it utilises component driven programming. A simple example of this paradigm are the arrows shot by the player in the game. In an inheritance based paradigm I would have developed a superclass called *Arrow* and then a subclass for each type of arrow; *fire*, *ice*, *poison* and *explosive*. However, as part of a component driven development process I implemented an *Arrow* class which has a variable storing any one of these effects as a component object. A benefit of this kind of software development can be seen when looking at the games arrow upgrade functionality. In the game it is possible to take a regular arrow and, through some interaction with the environment, convert it into an elemental arrow. In the inheritance paradigm, when the player upgrades an arrow, I would have had to first destroy the old arrow and then instantiate a new elemental arrow in the same location as the previous arrow. Using component driven programming all I needed to do was attach the elemental effect object to the existing arrow, a far more elegant and seamless solution. It was a somewhat steep learning curve to carry this paradigm across my entire code base, but it made my code easier to work with in the long run.

Finally, it is worth noting that version control was very useful in my development cycles. I used Unity's own version control implementation called *Collaborate* (Figure 2.1). It's a simpler system than the ubiquitous version control systems git and mercurial. It allows for commit operations, pushing and pulling to and from unity's own remote server and reverting to a previous build. It does not support features like branching, but as I was working on my own this was not an issue. It gave me the security of rapidly prototyping ideas on top of the current build and reverting it whenever something went amiss or the idea was not worth pursuing.

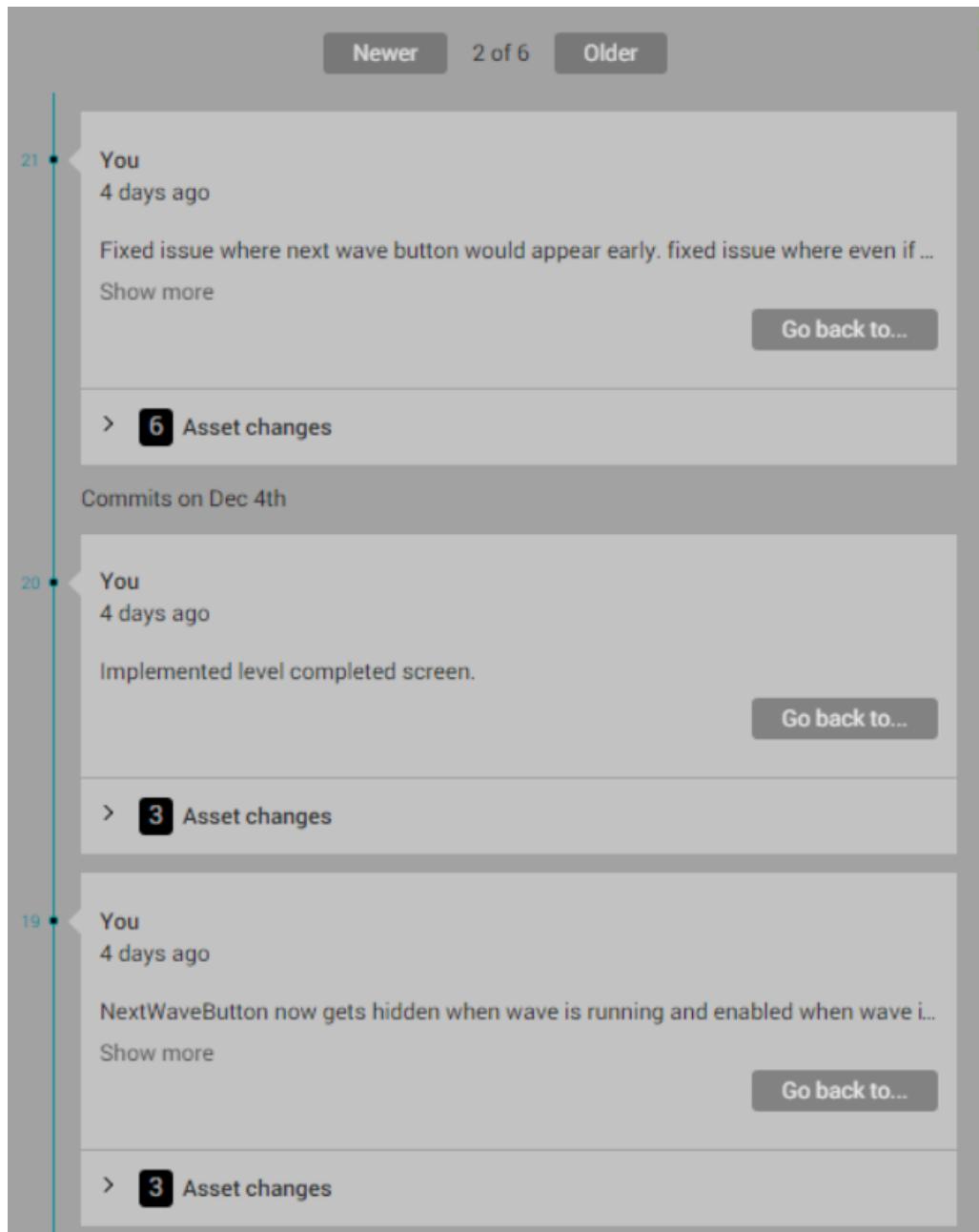


Figure 2.1: Unity Collaborate displaying three of my commits.

Chapter 3

Game Design

3.1 Game Mechanics

3.1.1 Creeps

Creeps differentiate from each other in a number of ways. First and foremost creeps have different travelling speeds, health values and gold rewards. They may also have various elemental weaknesses and spawn times. Some special creeps such as the lich and the dragon boss can shoot projectiles at the players. In total, there are only 5 graphical representations for the creeps due to the financial cost of procuring good 3D assets. Because there are many more ways to configure the creeps in code than there are visual representations, attributes such as health and speed for a given creep may vary across the waves. However the core characteristics of each creep remain the same and are as follows:

Lich: The lich creep is usually spawned in small numbers as they are very powerful. In comparison to other mobs they have a moderate health pool and moderate speed. They are able to conjure magical projectiles that they can hurl at the player. The player is given a brief time frame to prepare for the attack as the lich creep undergoes its preparation animation. On completion of the animation the projectile travels at a constant velocity towards the player. If the player gets hit they lose a life so they must physically move their head in order to avoid



Figure 3.1: The spawning locations are marked by red circles and the target location is marked by a green circle

the projectile. It is unwise to turn your back on these creeps as you can get hit by surprise and it is usually a good idea to kill these creeps first.

Skeleton Warrior: Skeleton warriors are usually spawned in rapid succession. They have a low health pool and travel at great speed. The players best tactic is to employ spirits that shoot forth explosive projectile that have an area of effect damage. Moreover, the player should use ice spirits to slow them down.

Death Knights: Death knights spawn in average numbers and in a moderate frequency. They have high health pools and travel slowly. It is best to employ poison spirits as they deal damage over time in percentage points of the overall health pool.

Death Riders: Death Riders are tough to kill as they have both a high health pool and a fast travel speed. Depending on the wave number they may spawn in low or in high numbers. The player should employ a combination of all spirits to take down these creeps. Use ice spirits and shoot ice arrows to slow them down while poison and fire spirits slowly wear them down.

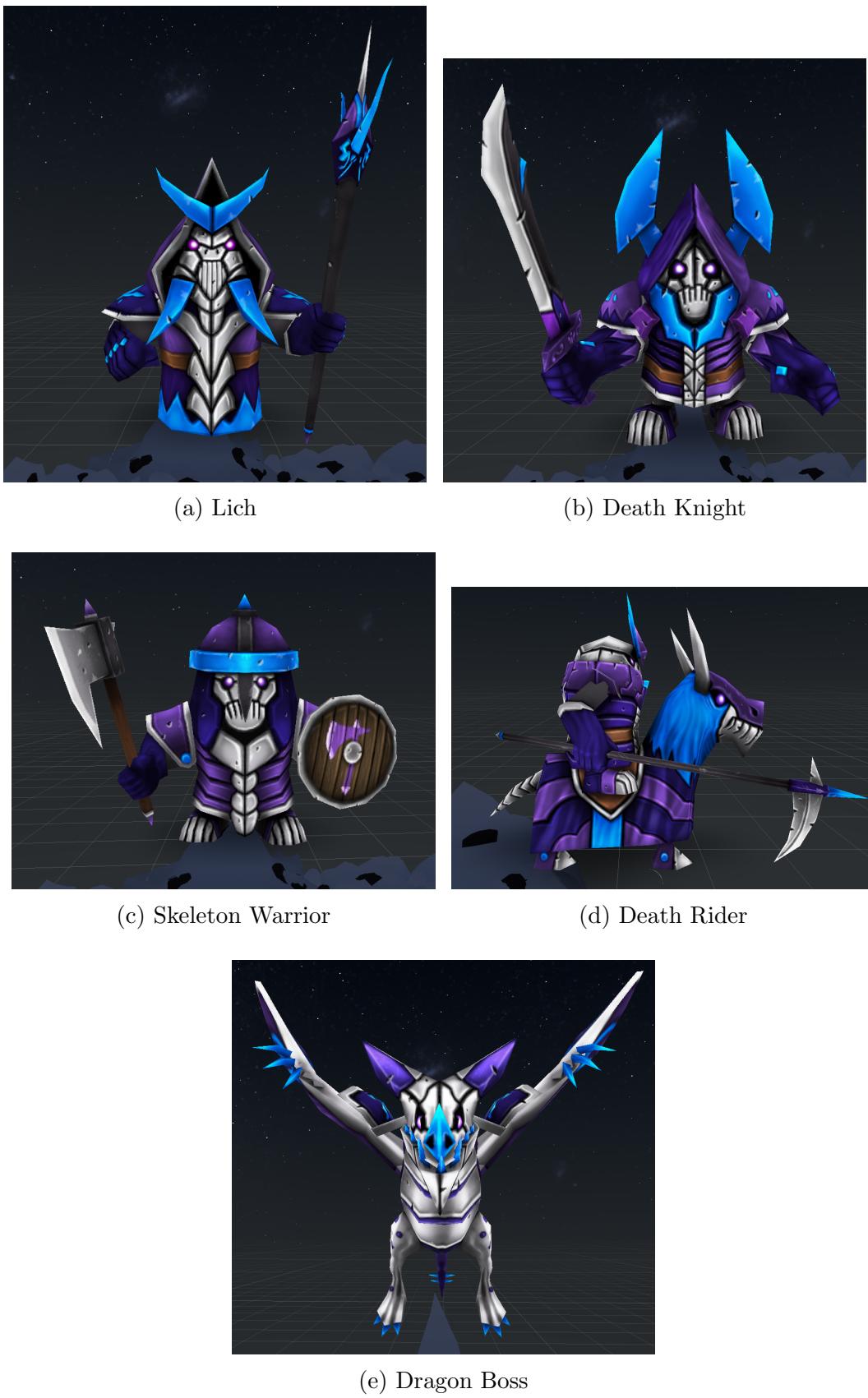


Figure 3.2: All the creeps

Dragon Boss: The dragon boss spawns in the last wave of a level. It is much larger in physical size than the other creeps and carries a much bigger health pool. It travels slowly and has a strong resistance against the poison debuff. The dragon boss also shoots forth huge magical projectiles that travel at a constant velocity towards the player. These projectiles are much bigger than the projectiles summoned by the lich creatures and the players only way to escape them is to teleport to another tower altogether. To defeat the dragon, the player should have constructed a spirit on every tower and have saved together some coin to buy explosive or fire arrows to bring the beast down before it reaches the temple.

3.1.2 Bow and Arrow

The bow is a modified version of an open source bow provided by Valve themselves. Valve provides the source code of the bow which is many 100s of lines of code and includes the bow and arrow graphical resources. The bow physics works incredibly well and I saw no reason to reinvent the wheel when a great solution already exists. The physics are tweaked slightly to provide the bow with a longer range and extra functionality was written to allow the bow to draw and shoot elemental arrows that have additional game logic attached to them. Moreover arrows have their damage scaled to the distance they have travelled. This is not realistic, but it makes shooting arrows from further away more rewarding for the player allowing them to hone their archery skills.

3.1.3 Towers

Towers are a perfect match for room scale virtual reality. Room scale virtual reality provides the user with a few square meters of space to walk around. The towers tops in the game only measure a few square meters in surface area themselves. This means that the player will encounter the immersion breaking HTC Vive chaperone barrier a lot less. Moreover it provides the player's brain with a tangible reason as to why they can't walk more than a few meters in a given direction. However, in order to not constrict the player too much and let them explore



Figure 3.3: A tower in it's highlighted state

the entire map, many towers are placed within each level. A player can press the trackpad button to activate a laser pointer and point at any tower in their field of vision. The tower will highlight in a blue colour to indicate the selection (Figure 3.3). When the player releases the trackpad they are instantly teleported to the other tower. Sometimes a tower may not be visible from the players current position, so multiple teleports may be required to get from one end of the map to other. Tower are scaled to different heights and placed at varying distances from the path to keep the player engaged. This design decision forces the player to think carefully about what spirits to assign to a tower.

3.2 Top Level Software Design

Figure 3.4 illustrates a top level perspective of how the game is implemented. The Game Controller keeps track of the core variables involved in the game play, such as the price of every spirit and elemental arrow and will handle player transactions. The Game Controller is also

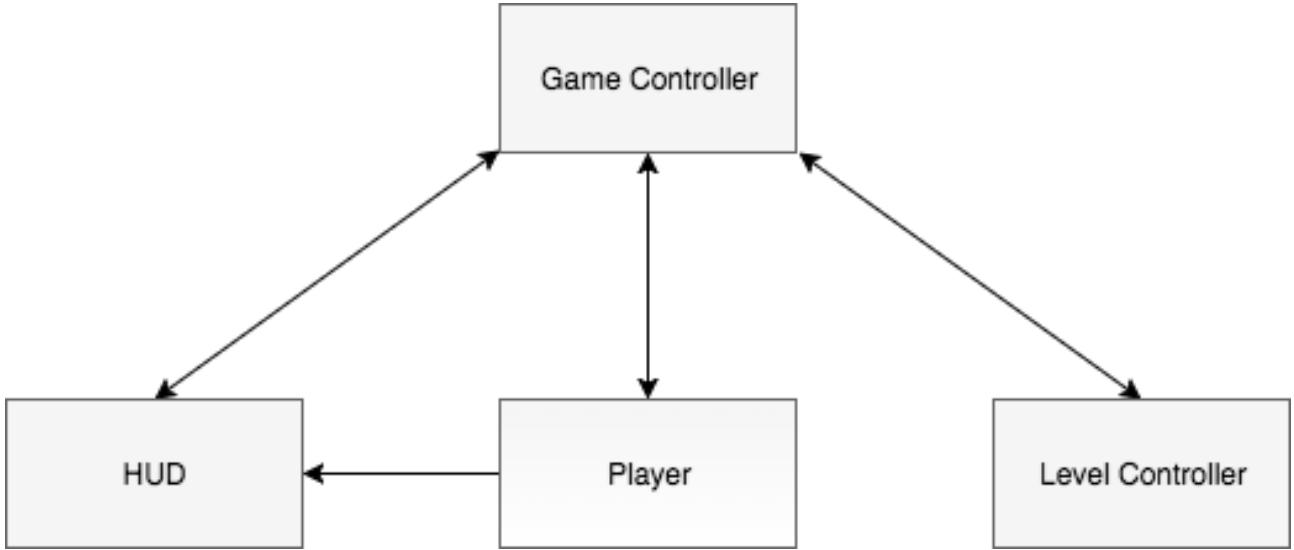


Figure 3.4: Information Flow Diagram representing a top level perspective of the system

responsible for operating the main game loop, such as checking if the players lives are depleted and signalling a game over event or triggering a game completed event after all levels have been completed. Each level can have multiple waves which in turn can have multiple spawners that spawn a series of creeps. When all creeps originating from a spawner have been slain the wave is notified and marked as completed. When all waves have been marked as completed the level is marked as completed and the game controller is signalled.

3.3 Player

The player objects consists of three tracked elements; the headset and the two controllers. Each of these elements can move independently of each other. The players headset represents the location of the main camera used to render the game. It also has a hit box attached so that things may collide with the player. The two controllers are used to interact with the world. The player can command the bow using one controller for the arrow and one controller for the bow, or they can use a controller as a laser pointer to teleport to another tower or interact with the HUD. The laser pointers are activated by pressing the trackpad and by default extends in the direction of the forward vector of the controller. Because the resolution of the headset is poor it looks like the laser pointers extend into infinity even though only the first 20 meters are

rendered. The laser ray cast extends 600 meters in the forward direction. If the laser intersects with a button in the HUD (Figure 3.5) or a tower that is closer than 20 meters, the rendering distance of the laser will shorten to the intersection point of the tip of the laser giving the effect that the player is selecting something.

3.4 **HUD**

The heads up menu (Figure 3.5) is displayed whenever the user presses the grip button on either controller. It can be deactivated by pressing the grip button once more. The HUD appears in the centre of the player's vision, overlapping the rest of the game's content. The HUD is used to display the players gold value and number of lives. Additionally, a player can buy one of the 4 spirit types from the list of buttons and start the next wave. In early stages of the development, the players lives and gold value were permanently overlapped on top of the players vision as they would be in a normal computer game. However this broke some of the immersion of being inside the game world. The downside of the current design is that the user may loose track of the number of lives if they are not careful. Initially, the HUD was static, but play testing with my supervisor showed that the lack of feedback from the HUD was confusing. To that end button highlighting was implemented. When a player hovers their laser pointer over a button it is slightly increased in size. Additionally, when the player clicks the next wave button it is hidden from view until the current wave is defeated. This allows the player to be certain that they pressed the next wave button and that the enemies are on their way.

3.5 **Graphics**

The graphics of the game are stylised in what is known as *low poly art*. This style is very popular for two reasons. Firstly, as is evident by the name, low poly art assets have a low count of polygons reducing the computation cost for the rendering of such models significantly. This lends itself very well to virtual reality as the game needs be rendered twice for each frame at



Figure 3.5: The heads up menu overlapping the rest of the game. The laser is currently highlighting the ice button

a frame rate of at least 90HZ to prevent motion sickness. Secondly, low poly art is easier to design and therefore cheaper to buy.

The art assets in the game can be broken down into the mountain and lake background scenery, the monsters, the decorations, HUD elements, particle effects, skybox and bridge tiles. The mountain lake, monsters, decorations and skybox were bought from the Unity asset store. The particle effects were also purchased from the asset store but required appropriation to match the game requirements. The HUD elements are being used free of charge from Flaticon [8] with appropriate credits. The bridge tiles were designed in collaboration with a freelancer working on Fiverr [9]. There were very specific requirements in place so that the bridge tiles could connect seamlessly after being placed by the path generation algorithm. More on this topic in the implementation chapter.

Chapter 4

Game Implementation

4.1 Path Generation

4.1.1 Development of the Algorithm in 2 Dimensions

In early prototypes of the game, I used an environment from the graphics asset store 4.1 with an immutable path. For a very brief period, I tried using my own 3D modelling skills to alter the path of the map, but quickly hit a brick wall in regards to how tedious it would be to generate different path layouts. My supervisor and I agreed that a procedural solution would be the best way forward. I used a graphics framework called "Processing 3" [10] to prototype the algorithm in 2 dimensions using the programming language Java. I built the generation algorithm in incremental steps introducing new parameters at each stage to shape the path that was generated into something that is visually appealing and challenging in terms of gameplay. Figure 4.2 showcases a path built by the finished algorithm. See <https://i.imgur.com/VoesXAE.gif> for an animation showcasing the algorithm building the path step by step.

The algorithm runs on top of a $n*n$ dimensional grid where each cell of the grid can either be a path or empty. In the first iteration of this implementation process I tried to apply a simple random walk that has an equal probability of walking in any of the 4 compass directions. Every

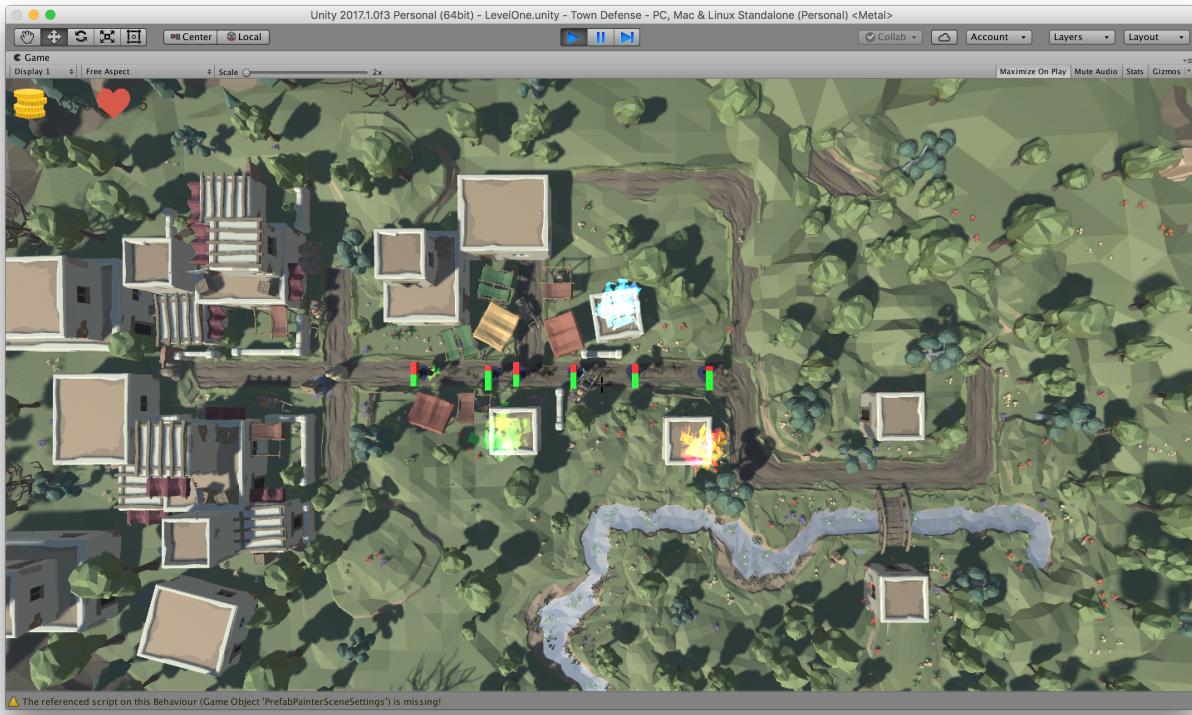


Figure 4.1: Prototype map with path seen from birds eye view

step of the random walk consists of a *corridor* - a series of successive cells running in the same direction. The length of a corridor in terms of cells is determined by a random number generator that is bounded by an upper and lower bound which are determined by input parameters. A path as a whole consists of a series of adjacent corridors. The number of corridors in a path is also determined by an input parameter. The entire path is bounded by the grid dimensions. If the random walk reaches the edge of the grid, the algorithm will pick a direction and corridor length that does not lead the path outside of the grid bounds. Figure 4.2 is an example of this simple algorithm with a corridor length ranging from 5 to 10 cells and a path length of 30 corridors.

There are several things wrong with the paths generated by this algorithm. First it allows the path to run directly adjacent to one another. This is aesthetically unpleasing and bad for game design as it causes all the gameplay to concentrate in one part of the map and leaves no spaces to place towers in-between the paths. Secondly, not depicted in figure 4.3, it allows paths to intersect in random nonsensical ways. Tower defence games require a path with a flow of direction, like several streams leading into one river. If the algorithm is allowed to

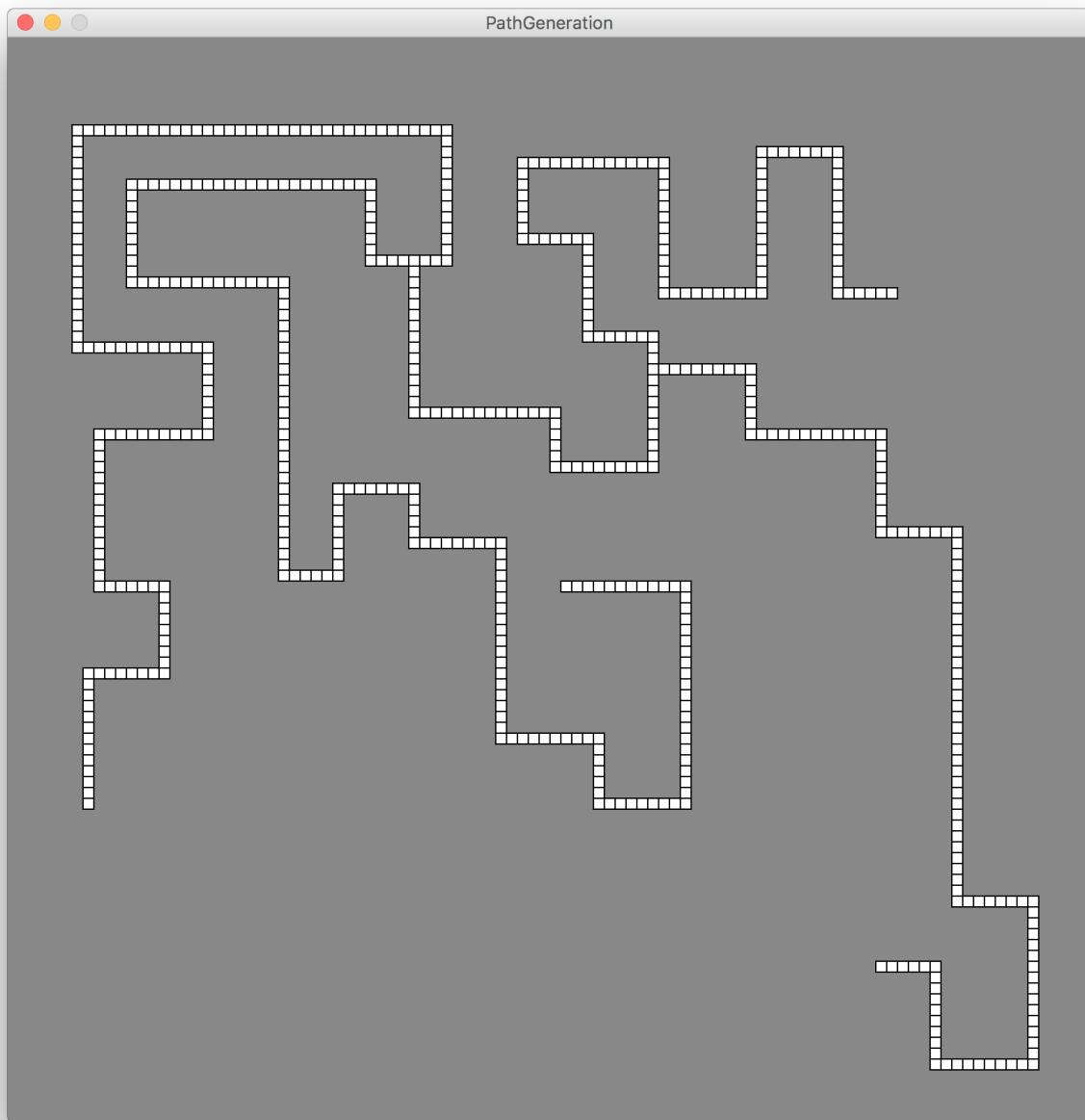


Figure 4.2: This path was generated with corridor length of 5-10 cells and a pathMargin of 4. The minimum and maximum branching coefficients are 0.3 and 0.8 respectively. See <https://i.imgur.com/VoesXAE.gif>

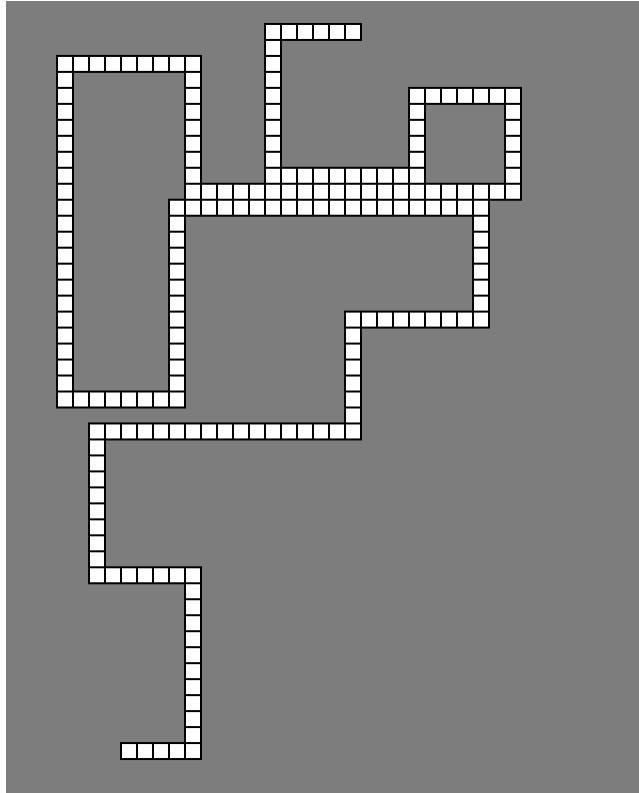


Figure 4.3: Unconstrained random walk with corridor lengths of 5-10 cells. Aesthetically unpleasing and bad design for gameplay

generate random intersections then there will be areas of the path that remain untravelled by any monster as it makes no sense for a creep to travel along a longer path than they have to.

To adjust for these shortcomings, I needed to introduce some limiting parameters into the algorithm. Namely, a strict margin that any one part of the path must maintain to another part of the path. This would prevent both the intersection problem and the parallel paths problem. Additionally, a margin must also be maintained to the bounds of the grid or else the path can exhibit a strong bias to one edge of the grid.

Introducing these restrictions to the algorithm has a clear downside. It can no longer be a simple random walk algorithm as the path generator would just get stuck when it fails the path margin constraint. As a consequence of this, I decided to introduce backtracking. The implementation of this is a recursive method that tries to generate a corridor of cells in a random compass direction. If the corridor generation in the chosen direction abides the margin constraint it is added to a stack of corridors representing the entire path. If the generation is unsuccessful, the algorithm randomly (it is important that successive compass directions

are chosen randomly or the path will have a bias) tries another of the 3 remaining directions. If none of the 4 compass directions result in a valid new corridor the algorithm pops the last generated corridor off the path stack and renews the search for a valid path with a new compass direction from the previous generated corridor. This has the backtracking effect you see in the animation provided for figure 4.2.

The recursive algorithm uses 4 helper methods for the generation of a corridor in each of the 4 compass directions. To explain how these helper methods work, I will demonstrate an example generation using south as the chosen direction and a path margin of 4. First the helper method generates the random length, l , of the corridor according to the lower and upper bounds provided. It will then attempt to loop l times and for each iteration the method will create a path cell to the south of the previous cell if and only if the following 4 conditions are fulfilled by the potential new path cell:

1. There are no existing path cells within a distance of 4 grid cells to south.
2. There are no existing path cells within a distance of 4 grid cells to the east.
3. There are no existing path cells within a distance of 4 grid cells to the west.
4. The edge of the grid is not within a distance of 4 grid cells to the new path cell.

If any of these conditions are violated we stop generating the corridor and return false to the main recursive method which will then try to generate a corridor in a different direction or backtrack if all compass direction possibilities have been exhausted. If a potential new cell in the corridor has been generated successfully we continue the iteration until l cells have been created and then add the entire corridor of cells to the path stack. Figure 4.4 illustrates a path resulting from this algorithm.

As you can see this algorithm only produces a path with two ends. In terms of game design this would mean that all enemies need to spawn in the same starting position. This would get boring very quickly. Part of the tower defence genre is to have to plan ahead to defend several branches of a path. Thus for every additional location we want to spawn enemies we need to add at an

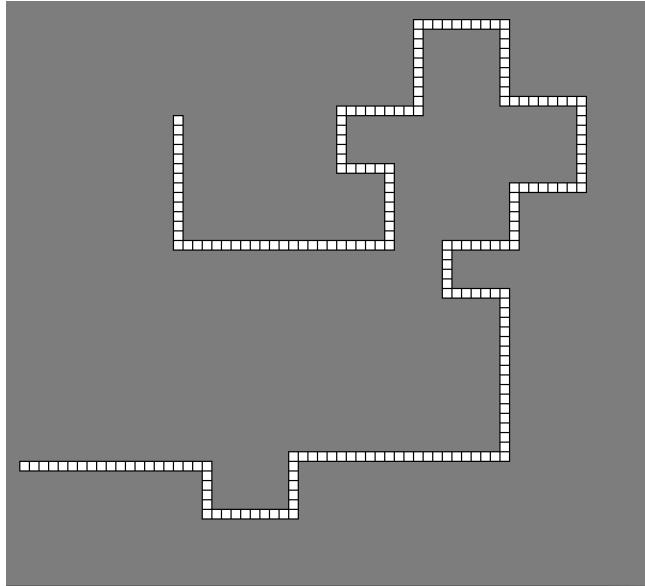


Figure 4.4: A Path with no branches and a length of 30 corridors, 5-10 cells in a corridor and a path margin of 4 cells

additional branch to the path. Using object orientated programming, I encapsulated the above described algorithm in a class called *RandomWalker* which takes the path defining parameters previously discussed in addition to a starting cell from which to build the path. It follows that all I needed to do was create a RandomWalker to build the root path and then create an additional random walker for each branch. The branching random walkers are instantiated with an initial cell that lies on the root path. Two sets of branching bounds were introduced to make the branches appear like a meaningful divergence from the original path and not just an extension of it; A minimum branching coefficient and a maximum branching coefficient. These bounds simply dictate that the branching random walker can start their randomwalk from any path cell on the main path that lies in-between cell number $\text{minBranchingCoeff} * \text{corridorLength}$ and $\text{maxBranchingCoeff} * \text{corridorLength}$.

Finally, it must be noted that the number of corridors could have been determined randomly for every branch, but it was determined most visually pleasing when every branch has the same number of corridors. Moreover, to introduce a greater variety between path generations it is possible for a random walker to branch off of a previous branch instead of the just the root path.



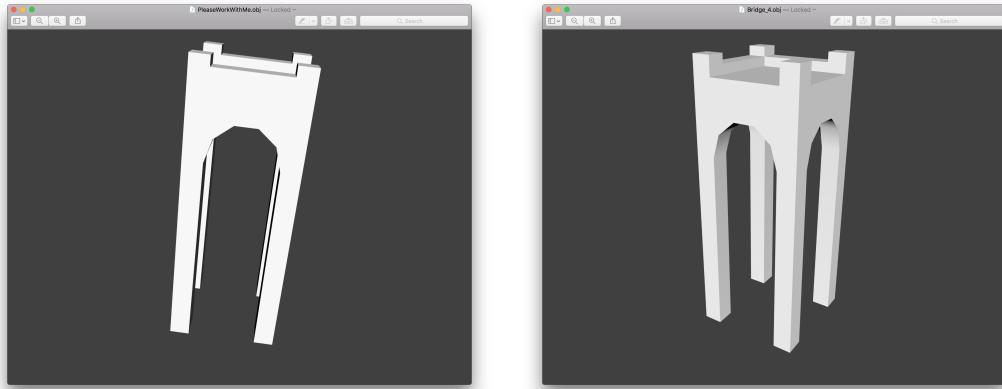
Figure 4.5: The spawning locations are marked red circles and the target location is marked by a green circle

4.1.2 Utilising the Algorithm inside Unity

After the algorithm displayed behaviour I was happy with, it was translated from Java into C Sharp to work inside the unity framework. A drawing algorithm was written to translate the generation algorithm's 2D output grid of cells and corridors into a 3D path in the game scene. There are 4 ways of rendering a given path cell as depicted in figure 4.6.

Each of the bridge models were created in collaboration with a freelancer working on the Fiverr platform [9]. I described exactly how I needed each of the bridge tiles to look like using a series of rough sketches and he converted my sketches into actual 3D models. It is important to note that the bridge art assets used in the game have mesh faces removed from any surface that will connect to another bridge. This reduces the computation costs of rendering the scene because significantly fewer polygons need to be computed.

The drawing algorithm iterates over all the cells in the grid of cells output by the algorithm and tries to draw any cell marked as a path cell. For each cell iteration the algorithm checks



(a) A straight bridge connected to two other path cells on either side (b) A corner bridge connected to two other adjacent path cells.



(c) An end of path bridge connected to only one other path cell (d) An intersection bridge connected to three other adjacent path cells.

Figure 4.6: All the ways of representing a path cell

how many neighbouring path cells exist and considers in which direction they lie in relation to the cell being considered.

- 1 neighbouring path cell must result in a straight bridge tile. The orientation must be adjusted to be in the direction of the neighbouring cell.
- 2 neighbouring path cells could mean we must draw either a straight bridge or a corner bridge. Therefore we check if the neighbouring cells are adjacent or opposite.
- 3 neighbouring path cells must result in an intersection bridge. The orientation is determined to be in the opposite direction of the single neighbouring empty cell.

An editor interface (Figure 4.7) was written for the Unity editor that allows us to adjust all the aforementioned parameters and generate a path inside the game world with the press of a button. This tool allows me or someone else to very rapidly create game levels. Although I am confident that this algorithm generates an appealing path the vast majority of the time, some paths still remain more interesting to play than others. At this point in time the algorithm is not used at run time but only for aid in level creation.

4.2 Creeps

4.2.1 Path Finding

When the path is generated every path cell is assigned a *path cell number* that represents how many path cells have been generated previously. The first path cell is assigned the number 0. The creeps use this numbering system for their path finding mechanism. A creep will always move from a higher numbering path cell to its nearest lower numbering path cell. This means that when a creep reaches an intersection bridge with multiple branching paths it will choose to travel down the correct branch because it's marked by the decline in path cell numbers.

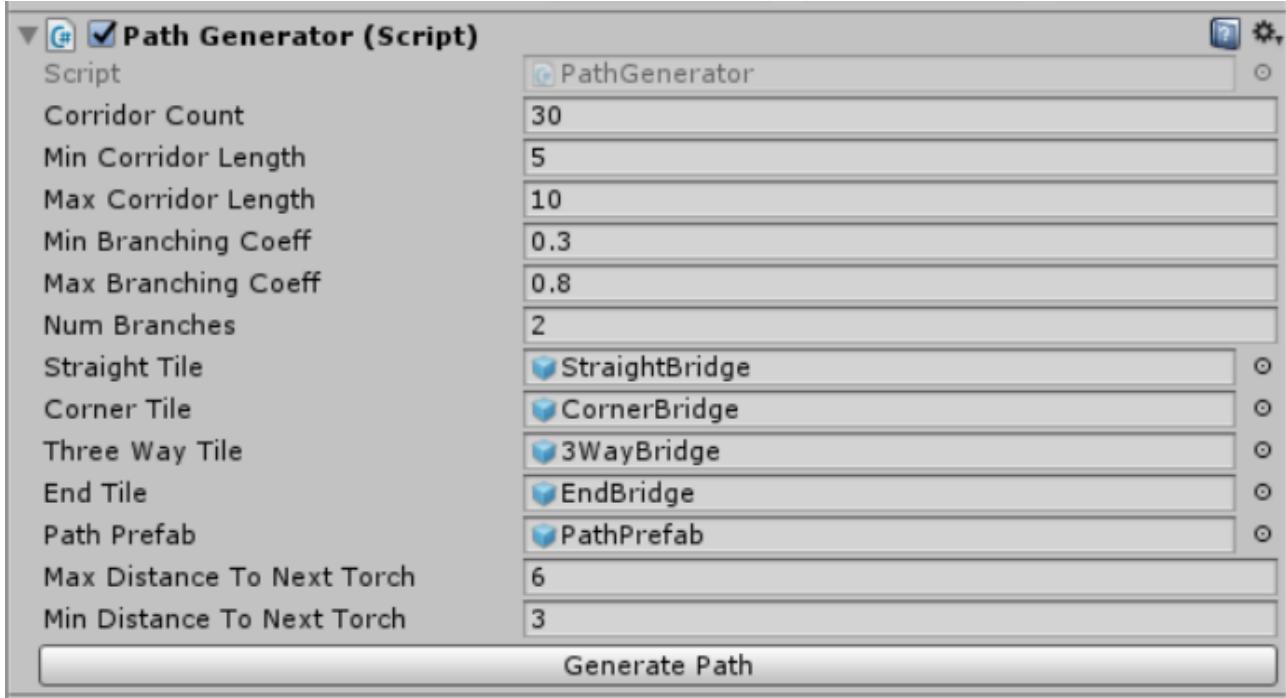


Figure 4.7: A path generation editor interface that allows for the generation of a path inside the unity editor

4.2.2 Spawners

The spawner script has a variable which references the bridge object (the graphical instantiation of a path cell) on top of which it should spawn its mobs. Several such spawners can exist across a map, each linked to a different bridge object. Every spawner stores an array of *CreepSpawn* objects. *CreepSpawn* objects store a *Creep* object which encapsulates a monsters core characteristics and visual representation. They also store a spawn time which represents the time to wait to spawn a creep, a movement speed value, a health value and a gold reward. When a spawner script is activated by its parent wave it will iterate over an array of *CreepSpawn* objects and instantiate the stored *Creep* object with the corresponding movement speed, health and gold rewards. Between each spawning iteration there is a time delay specified by the spawn time of the next *CreepSpawn* object in the array. The *CreepSpawn* class is serialised which allows the spawner script's array fields to be edited within the Unity editor as part of the the level designing process. Figure 4.8 illustrates the spawner editor inside the Unity editor.

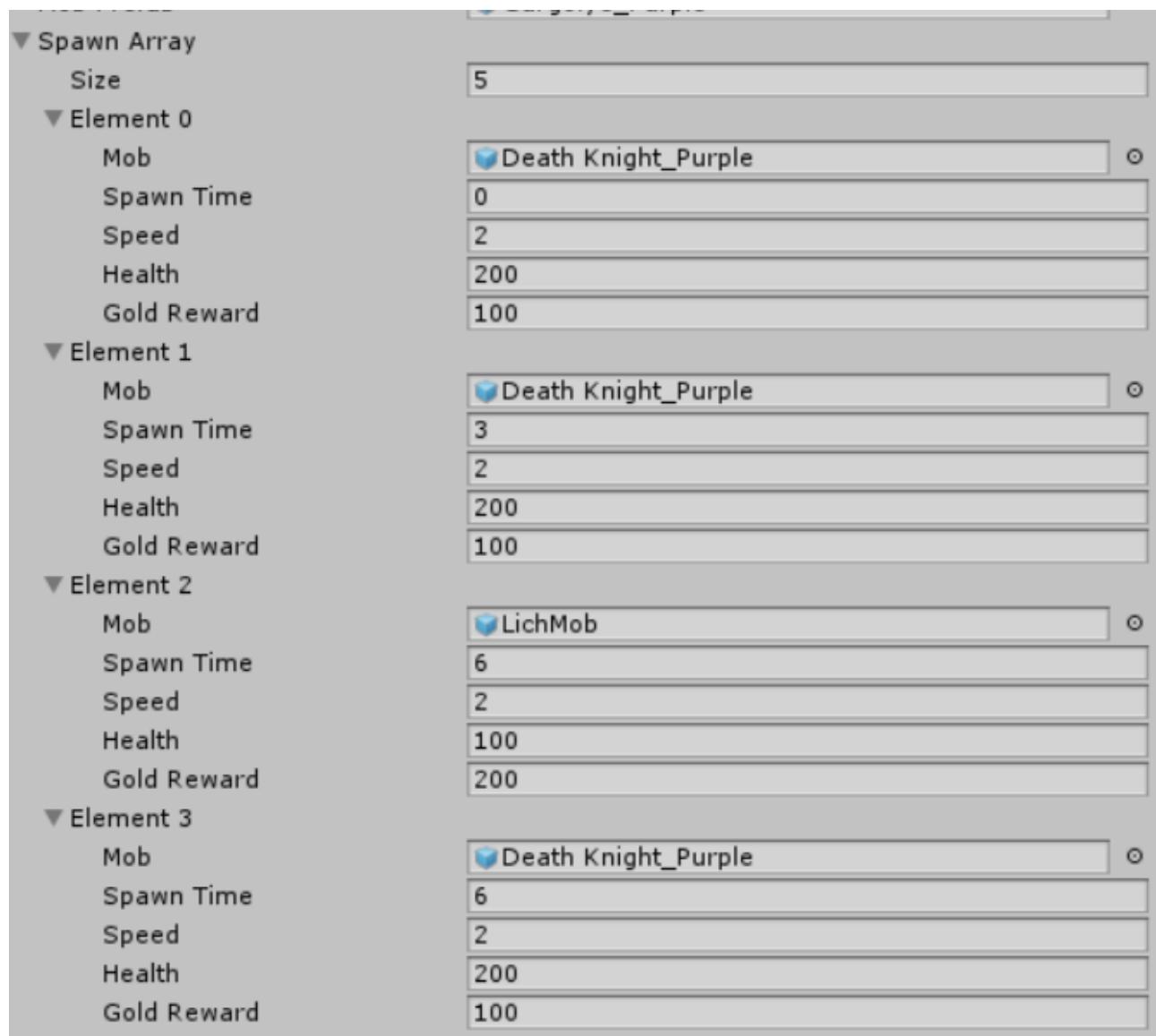


Figure 4.8: Spawn editor that allows the level designer to assign a sequence of mobs that will spawn at a given spawner.

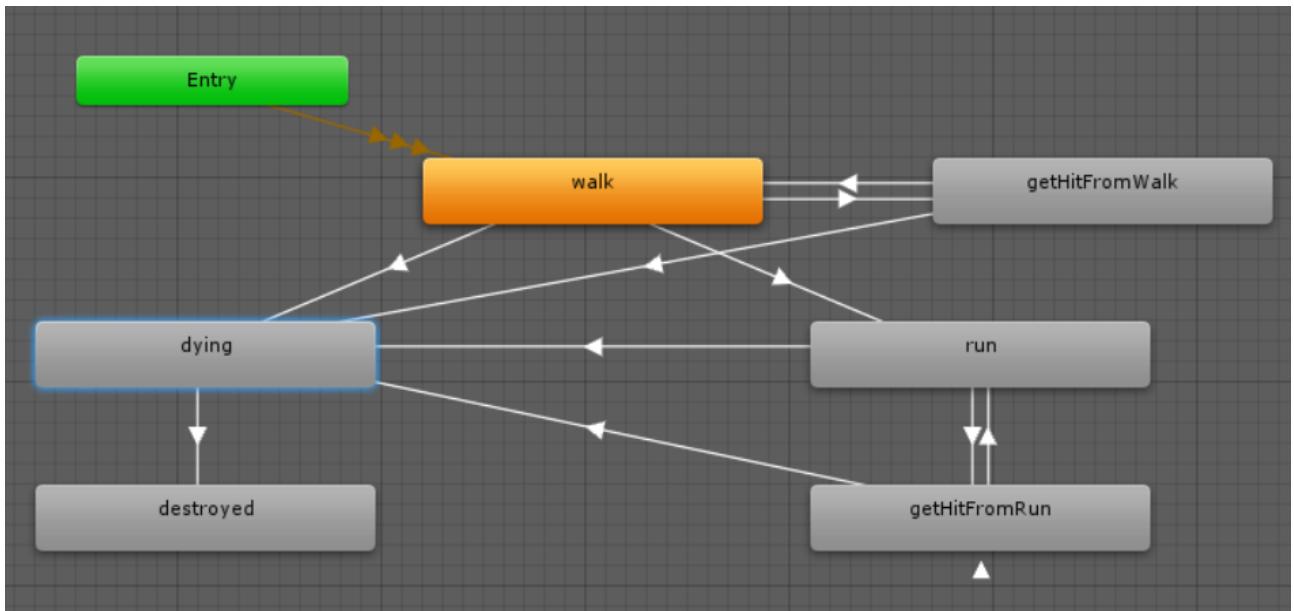


Figure 4.9: Death Knight animation transition system.

4.2.3 Animation

Creeps are animated using a transition system. An example of which can be seen in figure 4.9. These systems have their own variables such as an integer representing the speed of a creep and a boolean called *isDead* indicating if the creep is dead. A creep starts in its entry state which by default is defined as *walk* in the transition system. If the speed variable exceeds a certain threshold the transition system will transition its current state to the *run* state. Each state represents the current animation the creep is exhibiting. Additionally, the transition systems can trigger events in other parts of the code base. For example, when a creep reaches 0 health we don't want it to be removed from the game scene straight away. It should first play its dying animation. Therefor when a creep reaches 0 health, the creep class will notify its animation system by setting the boolean value *isDead* to true. This will force the transition system to move its current state to *dying* which will in turn trigger the creeps dying animation. On completion of that animation, the transition system will notify the creep object that the animation has completed and the creep object will remove itself from the game.

4.2.4 Shooting at the Player

Some creeps such as the lich and the dragon can shoot projectiles at the player. Inline with a component driven implementation, this ability is programmed on top of the regular creep implementation by attaching an extra script called *lich* or *dragon* respectively to a regular creep object. These scripts take the following parameters: *randomAttackTime*, *cooldownTime*, *projectileRange* and *projectileSpeed*. The *randomAttackTime* specifies the probability of the monster attacking in the next time unit. If the *randomAttackTime* is 0.2 then there is a 1/5 chance that within the next second the monster will attack. Once the monster has executed an attack there is a *cooldownTime* measured in seconds in which it can not attack again. The *projectileRange* is simply the maximum distance from which the creep can shoot the player. The actual shooting calculation is performed as follows: The position of both the player and the monster is ascertained. These two position vectors are subtracted from one another and then normalised. This give us the unit vector representing the direction in 3D space of where the creep should shoot. Finally, we multiply this unit vector by the *projectileSpeed* to obtain the velocity of the projectile. We then instantiate the projectile above the creeps head and apply the calculated velocity onto the projectile's rigidbody.

Chapter 5

Evaluation

5.1 User Testing

The user testing consisted of participants playing the game and subsequently answering an open ended survey attached as Appendix A. The survey was conducted in a free form fashion because of two reasons. Firstly, the participants were sourced from an enthusiast forum revolving around HTC Vive games. This meant that the participants were aware of the usability requirements of virtual reality and could give valuable insight in the actual game play. Secondly, by its very nature creating a computer game is a creative process and open ended survey questions allow for creative responses.

There were a total of 12 anonymous participants sourced from the subreddit community r/vive [11]. Each participant was shown a game manual, attached in Appendix C, before playing the game. The survey responses are attached in Appendix B. The distinct advantage of sourcing participants from this community rather than my fellow St Andrews students is that the majority of the population has not tried a sophisticated VR experience such as the HTC Vive. This would risk obscuring the data collected by a players bias to be impressed with virtual reality itself rather than my game. To evaluate this feedback I will examine common patterns found in the feedback and illustrate what has been done to address the feedback.

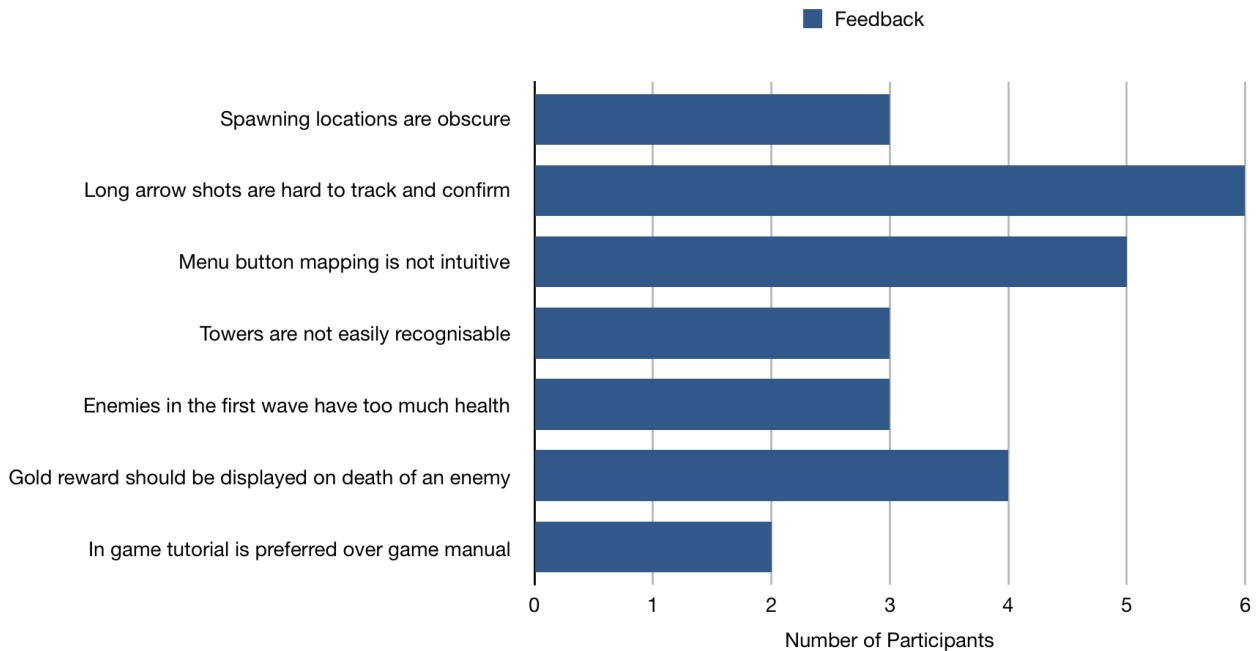


Figure 5.1: Graph depicting number of players that responded with the feedback displayed on the Y axis.

In total there were 7 responses that more than 1 person mentioned. They are depicted by graph 5.1 and listed bellow.

5.1.1 Spawning locations are obscure

Initially my intention was to surprise the player with where the monsters are spawned. However, several participants voiced their displeasure with this design decision. They preferred to know ahead of time where the monsters will come from so they can plan and build relevant towers.

"It was never clear where the monsters were going to be coming from or where they were going"
- SearangerX, Appendix B

To address this issue I implemented purple rays of light highlighting the monster spawn points (Figure 5.2). I also highlighted the destination temple of the monsters in a coloured light.

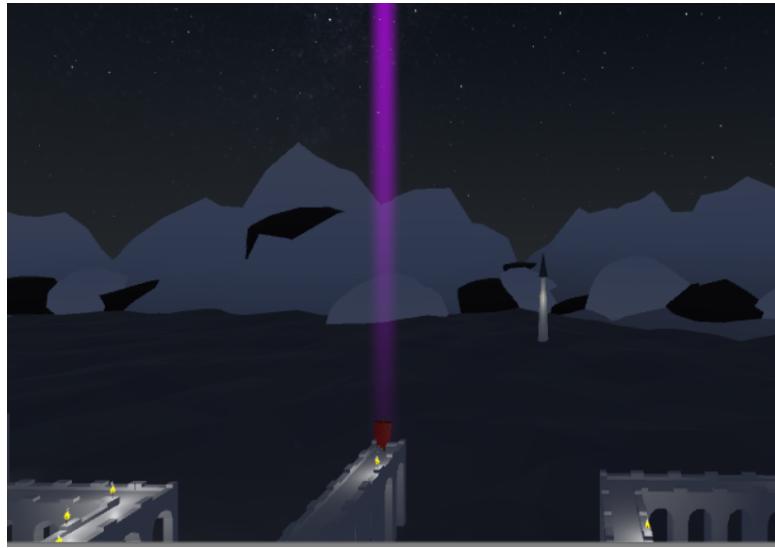


Figure 5.2: Purple ray of light indicates the bridge where the enemies will spawn

5.1.2 Long arrow shots are hard to track and confirm

Because the resolution on current generation VR headsets is very low it makes it more difficult to see the arrows as they travel. Moreover, when a player is firing at enemies alongside the spirits it can become difficult to tell whether the enemy took damage because the arrow hit home or because the spirit hit the monster. The participants requested some form of audio feedback whenever the player hits an enemy with an arrow.

"A hitmarker sound for when you hit an enemy with your bow would be satisfying" - Camo41,
Appendix B

To address this issue I extended the time the arrow glint particle effect stays on so the player can more easily track the arrow and introduced an audio clip that is played whenever the player successfully hits an enemy with their bow and arrow.

5.1.3 Menu button mapping is not intuitive

This feedback is due to a very obvious oversight of mine. The HTC vive controller actually has a menu button and I mapped the menu toggle to the grip button, so participants were confused on how to summon the menu.

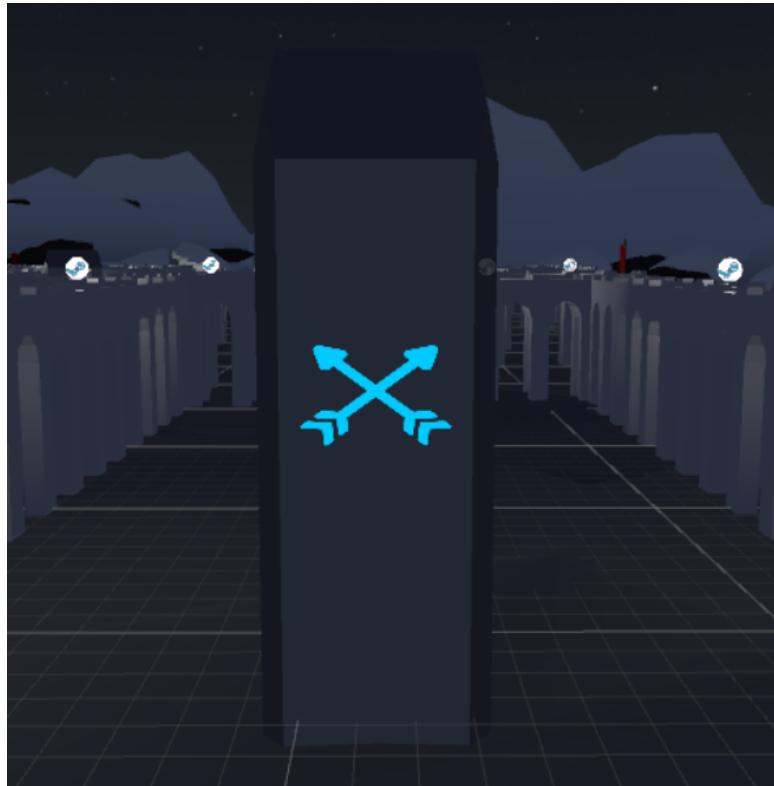


Figure 5.3: Illuminated blue arrow icon indicate that this an archery tower

"Grip to menu was odd, but it worked. I'd suggest using the menu button on the Vive for the menu, but after about 5 seconds grip for menu just clicked and it worked out fine. More of a personal taste thing, really. Otherwise, it's straightforward bow use and menu pointing, no point reinventing the wheel here." - Herballistic, Appendix B

I changed the menu button.

5.1.4 Towers are not easily recognisable

The graphical representation of the towers were not obvious enough to the participants. There was no illumination and it was hard to see where the other towers in the map resided.

"The towers should stand out more, specially when inhabited by a spirit." - rodwin0, Appendix B

To address this concern I applied an arrow icon to each face of a tower. The icon is self illuminated in blue and makes it easy to see a tower from across the map. See figure 5.3.

5.1.5 Enemies in the first wave have too much health

Whilst making the game, over the course of many months, I developed some skill at hitting the creeps with the bow and arrow. I am particularly good at wave 1, because that is the first wave I test new features on. This tendency gave me a bias to make the first wave much harder than it should be for an average player.

”The monsters seemed to soak up a lot of damage and the first round with no towers was very tedious.” - Searangerx, Appendix B

To remedy this fault I lowered the health of each monster in the first wave and increased the gold reward.

5.1.6 Gold reward should be displayed on death of an enemy

Money is rewarded to the player for each monster they kill. The reward is proportionate to the difficulty of killing the monster. In response to the question *”Do you feel like the game rewards you for progressing through the waves of enemies?”* some participants said that they did not feel rewarded because they were not able to actively track the amount of gold they were being rewarded.

”If you have the menu open, yes, because you can see the gold going up. I would suggest having a graphic over an enemy you just killed showing how much gold you just earned” - m0jamb0, Appendix B

Time did not allow for this feedback to be addressed, but a future implementation of the game would see a graphic appear above a creep’s head indicating the gold reward when the creep dies.

5.1.7 In game tutorial is preferred over game manual

Even though only two participants explicitly mentioned that they did not read the game manual and would have preferred a tutorial, it was obvious that several other participants had also learned how to play through trial and error. Game manuals are a thing of the past and players are used to being taught how to play by the game.

"I would definitely start the game out with the menu turned on with directions on how to use the controls. It took me a good few minutes to mess the controllers and the towers to figure out how to even move between towers" - Camo41, Appendix B

Time did not allow for this feedback to be addressed, but in a future implementation of the game I would implement exactly what the participants suggested: having the menu open to start with and displaying text instructions on how to play the game.

5.2 DOER Requirement Comparison

This section will look at the initial DOER objectives (See Appendix D) and how they have evolved throughout the project. A comparison will be made between the final game and the evolved objectives.

5.2.1 Primary Objectives

Implement a basic tower defence game in a 3D virtual space

This requirement has remained the same. The finished game operates in 3D virtual space and has several waves of enemies that build up to a concluding wave with a boss fight. This requirement has been achieved.

Implement a tactical birds eye view of the game where the player can strategically place towers and upgrade existing towers.

This objective has been omitted after testing revealed a general sense of unease experienced by hovering high up in the sky without solid support beneath ones feet. A tactical view of the game map can now be observed from one of the tall towers in the game. Towers have different heights on purpose so that a player must teleport to one of the higher tower to get an overview of the game play.

Implement a first person view of the game where a player can engage in ranged combat from within one of the towers.

This objective has remained unchanged and has been implemented fully, utilising a teleportation functionality with which the player can navigate the game's towers and a bow and arrow with which they can shoot down enemies.

5.2.2 Secondary Objectives

Implement challenging AI used by the enemies to cross the map.

The creeps utilise basic path finding to cross the map and the lich and dragon mobs use AI to determine when and how to shoot the player. Future development on the game would see more special enemies like the lich. For example, a healing creep that would occasionally heal other nearby monsters.

Implement a variety of interesting weapons for the player to fight with

There was a prototype that allowed the player to throw objects at the enemies, however, it quickly became clear that this mechanic is not accurate enough and it was very tiring for a player's arm. An alternative objective was formulated - players can convert their arrows into

elemental arrows for versatility. This way a player can strategically shoot fire, ice, poison and explosive arrows. This turned out to be very beneficial to the tactical side of the game. For example using ice arrows a player can specifically target more dangerous or faster enemies in order to slow them down.

Implement a reward system where players can earn rewards that persist through game rounds

This objective remains unchanged and has been achieved by implementing a gold currency in the game whereby each enemy killed provides a reward in gold coins proportionate to how difficult it was to kill that enemy. This currency can then be spent to purchase spirits or elemental arrows.

5.2.3 Tertiary Objectives

Create multiple maps/levels

Only one level with 5 waves has been created. However, an additional objective was introduced. A procedural generation algorithm was written that creates a random path layouts for the monsters to follow, thus facilitating the design of further levels. The procedural algorithm produces satisfactory results most of the time, but some generated paths still remain somewhat more interesting than others and therefore they are not generated at run time. Further development of the game would include making the game entirely procedural such that paths are generated at run time.

Create boss fights to make the game more diverse and exciting

This objective has persisted and a dragon boss fight was implemented in the final wave of the level. The dragon boss is visibly bigger and has more health than other creeps. Additionally,

the dragon has a special attack that shoots forth a projectile that a player can only evade by teleporting to a different tower.

5.3 Future Work

Future work on this game would entail making the game completely procedural, so that every level would be different at run time. Additional creep types would be introduced with different special abilities such as healing other creeps or teleporting across path branches. More spirit types would also be introduced with the possibility to upgrade a spirit into dealing more damage or even firing multiple projectiles. Different environments would be used to host the generated path such as the inside of volcano with lava all around the path. I would also address the remaining points of feedback that I did not have the time to address, such as providing visual feedback of the gold reward and introducing a tutorial at the start of the game. Finally, I would implement a mechanism that allows the player to upgrade his bow, allowing for more powerful shots.

Chapter 6

Concluding Statements

The project successfully implements a virtual reality tower defence game with a total of 5 waves and a boss fight. The player is able to fight in first person using a bow and arrow in addition to managing an array of elemental turrets. A procedural path generation algorithm and a suite of level design tools have been created to facilitate the creation of further levels. Even though there have been significant changes to the project objectives, the core concept of the game has remained true to its conception. For every change to the project objectives an appropriate justification has been provided.

Play testing revealed valuable insights into both the merits and drawbacks of the game. All participants were sourced from a virtual reality enthusiast forum which enabled them to incorporate knowledge of other virtual reality games into their feedback. The feedback of any given participant was often consistent with the feedback provided by other participants which made the evaluation of the game easier to conduct. Wherever possible, collective negative feedback has been addressed and appropriate changes have been made. If time were to allow it a second round of feedback would be conducted using the same participants.

Further development of the game would include making the path generation procedural at run time. Spirit and bow upgrades would be implemented to give the player a greater sense of progress. Additional environments would also be included to keep the player engaged. Several of the play testers indicated that, with the aforementioned changes, they would like to see the

game accessible on steam. Perhaps in the future this ambition will be pursued.

Bibliography

- [1] David Soos. History of tower defense games. <http://ezinearticles.com/?History-of-Tower-Defense-Games&id=7241055>, Aug 2012. [Accessed: 2017-09-25].
- [2] Nic Simmonds. Tower defense a brief history. <https://mygaming.co.za/news/features/6341-tower-defense-a-brief-history.html>, Jun 2010. [Accessed: 2017-09-25].
- [3] History of virtual reality. <https://www.vrs.org.uk/virtual-reality/history.html>. [Accessed: 2017-09-30].
- [4] Devindra. Oculus rift vs. htc vive: What we've learned after the reviews. <https://www.engadget.com/2016/04/08/oculus-rift-vs-htc-vive/>, Jul 2016. [Accessed: 2017-09-30].
- [5] What is room-scale vr (room-scale virtual reality)? - definition from whatis.com. <http://whatis.techtarget.com/definition/room-scale-VR-room-scale-virtual-reality>. [Accessed: 2017-09-30].
- [6] Unity game engine. <https://unity3d.com/>.
- [7] Jon Brodkin. How unity3d became a game-development beast. <https://insights.dice.com/2013/06/03/how-unity3d-become-a-game-development-beast/>, Jun 2013. [Accessed: 2017-10-05].
- [8] Graphic resources s.l. <https://www.flaticon.com/>.
- [9] Fiverr international limited. <https://www.fiverr.com/>.

[10] Processing graphics engine. <https://processing.org>.

[11] Reddit r/vive. <https://www.reddit.com/r/Vive/>.

Appendix A

Survey Questions

1. How challenging is the game? If it is too difficult or too easy please suggest why.
2. The map layout is procedurally generated. Does the generated path provide for an interesting experience?
3. Are the controls intuitive? Please suggest any possible improvements.
4. Do you feel like the game rewards you for progressing through the waves of enemies?
5. Are the graphical representations of game entities such as the towers, bridges and spirits easy to see and recognise?
6. Please suggest a feature that would make the game more fun and engaging.
7. What aspects of the game did you like?
8. What aspects of the game did you not like?

Appendix B

Survey Responses

Camo41

How challenging is the game? If it is too difficult or too easy please suggest why.

The games difficulty is actually pretty great the bow actually help when fighting the enemies but the towers are essential for crowd control.

The map layout is procedurally generated. Does the generated path provide for an interesting experience?

It does, although there wasn't any variation throughout the paths besides turns.

Are the controls intuitive? Please suggest any possible improvements.

I would definitely start the game out with the menu turned on with directions on how to use the controls. It took me a good few minutes to mess the controllers and the towers to figure out how to even move between towers.

Do you feel like the game rewards you for progressing through the waves of enemies?

Not necessarily, I feel as though there should be like a "wave finished" sound or something, as well as maybe earning a "super point" the permanently upgrade stuff like bow damage at the end of a wave. to make it like your working towards something

Are the graphical representations of game entities such as the towers, bridges and spirits easy to see and recognise?

Mostly, I would consider adding a coloured border or some flags on the towers for them to stand out a bit more like their the "good guys."

Please suggest a feature that would make the game more fun and engaging.

I would put in some more towers or upgrades, and also a hitmarker sound for when you hit an enemy with your bow would be satisfying. I would also suggest making a flash of light or an aura of colour of the tower you just bought appear, because after I bought a tower for the first time I was a bit confused until I looked up. I actually thought the spirit was mesmerising when right below it.

What aspects of the game did you like?

I like the jumping around towers to help defend certain parts of the map and whatever tower you were on is what tower you built the spirit on.

What aspects of the game did you not like?

Mostly the repetitive music and lack of other noises besides the bow. Also the lack of in game instructions. There was also a lack of variety around the map, it was blandish with grey and black. Which actually made the spirits colours really pop! I was also confused of whether i should duck and dodge the projectiles shot at you, or whether i just move towers instead. All in all i really enjoyed the aspect of your game very much.

Mnemnoth

How challenging is the game? If it is too difficult or too easy please suggest why.

At first it was pretty balanced on the side of difficult. Although on the second and third play through it got a lot easier as I could pick where the enemies would spawn and place turrets accordingly. It didn't seem to change procedurally in that the enemy spawns and level layout were the same each time which made it easier as I could anticipate what would happen. This is true to tower defense games though (I think) in that you can restart with extra knowledge of what to do?

The map layout is procedurally generated. Does the generated path provide for an interesting experience?

Firstly the map didn't seem to change when I restarted the level or restarted the game. Which is what I expected by procedural (may be a bug). If not though and you have reseeded until you were happy with the layout then the level presented was fine. The different paths seemed balanced in terms of 90 degree turns and the time it took the enemies to move from the spawn point to the end point.

Are the controls intuitive? Please suggest any possible improvements.

The shooting controls are fine quick to get a hang of and I could snipe characters from a few towers away with practice. The glow effect on the arrows once they are far away is great and helps the user to track the arrows, great idea! That said there was a limitation in that the glow effect of the arrows would disappear at a certain distance which would make it difficult to readjust unless I hit an enemy while sniping from really far away. Can you extend the cutoff for the arrow glow effect? For example if I shoot the arrow straight up it will disappear and then reappear due the render cutoff (Arrow is there and tracked but the glow disappears then reappears).

The menu button on Vive controllers should be the system button above the track pad (only because I am used to this in other games).

The grip button should be used to toggle picking up the bow (again for consistency with other games).

For the laser menu pointer. It works as is but I was expecting that if I pointed into the menu area the laser pointer would appear. Once within the menu area/plane I expected that the trigger button would select an option. Track pad works but again is different to I'm used to in other games. For reference look at 'Big Screen' and the way it handles menus.

Room-scale positioning: I use room-scale and when booted up the game had me facing the

computer. Normally when games booted in room scale it would be rotated 180 so i am facing away from the computer. I tried resetting seated position but this had no effect. (I am new to using room-scale - used to use seated so this may be my/user error).

Do you feel like the game rewards you for progressing through the waves of enemies?

Yes, by allowing me to buy more towers and place explosive towers in the high up towers. Side note: It seems the towers higher up have more range or at least that is what I thought. It would be cool if there was a circle showing the range of the towers before you placed them as I wasn't sure if the towers would hit and how far. EG Ice towers/ Poison towers. I feel that it was too easy to save money and buy the explosive towers mixed with ice towers. I found that I didn't imbue my arrows but instead just saved for high power towers while shooting with default arrows. Which may be intended game-play wise. Perhaps (and this may be too confusing for players) there could be two forms of currency. Less main currency for buying towers and some other form of currency or 'charges' for imbuing your weapon with the elements. To encourage the imbuing of your arrows rather than sit back and let the towers do the work (Full disclosure I am a turtler in RTS games so maybe just my style).

Are the graphical representations of game entities such as the towers, bridges and spirits easy to see and recognise?

Yes! Although some suggestions for clarity: Could you make the tower "T" symbol glow or be the colour of the fire on the torches? By the way it reminds me of the Tesla logo (In a good way haha). The blocks where the path turns 90 degrees do not have the same archway geometry underneath but instead are a square cutoff. Could these also have the archway underneath for consistency? The poison effect when an enemy is hit (I'm basing this on the cavalry) seems to be too low. Perhaps raising this effect higher to the riders for this enemy type would make it more noticeable?

Please suggest a feature that would make the game more fun and engaging.

Game play wise it is good. But audio feedback when I damage (quick soft sound) or kill enemies (loud "arrgghhh") would be great and make me feel more powerful/reinforce my performance. Perhaps only when the player arrow itself hits as sounds for each tower hit may get overwhelming or too numerous). Although this could also be hilarious hearing the multiple enemies getting hit. A damage tick sound for poison effect (Could be grunting by the enemies or a hissing gas sound? I dunno). A sound when an enemy dies and the player receives gold (Cha Ching!). A strong more intense Cha Ching! sound when the players money crosses the threshold and they can purchase a new tower. This means the player will know they can teleport and place another tower mid round rather than using to menu to check or waiting until the end of the round.

What aspects of the game did you like?

The enemy design and the look of the elements above the towers. The ease of moving from tower to tower (and the help in line of sight from being high up that the middle towers gave) although it didn't seem too necessary in the early waves. The ability to imbue your shots with the tower

elements which I didn't use as much as I should have. My fault haha. The feel of shooting the bow and arrow was tight and good and after a few tries I could snipe from far away. Very satisfying!

What aspects of the game did you not like?

Not much left from my notes. Although lack of hit feedback/player getting damaged. I couldn't tell if I lost a heart without opening the menu. If a 'shaman' hits the player would be nice to have a red blood effect and a sound on screen to tell the player they have lost a heart/life.

Footface_

How challenging is the game? If it is too difficult or too easy please suggest why.

The game was not too challenging, i guess on a scale from 1-5 1 being easy 5 being super hard, it was a decent 2.

The map layout is procedurally generated. Does the generated path provide for an interesting experience?

Yeah that was pretty cool, altho in a full tower defence game i would prefer designed maps with varying difficulty.

Are the controls intuitive? Please suggest any possible improvements.

The controls were ok, I don't have any suggestions on how to improve them.

Do you feel like the game rewards you for progressing through the waves of enemies?

Not really, I mean its cool that you get different amounts of gold for kills based on the HP of the enemies, but I just felt like the progression wasn't really there, I mean yeah I got spirits up and all, but i guess it would be cool with more spirits and maybe a way to upgrade them?

Are the graphical representations of game entities such as the towers, bridges and spirits easy to see and recognise?

Yeah, that was easy enough, would be nice to maybe get a visual representation of what spawn points were going to be used before starting each wave.

Please suggest a feature that would make the game more fun and engaging.

A way to upgrade the towers would be nice, maybe be able to combine elements? Maybe add a lightening element that bounce between enemies.

What aspects of the game did you like?

It was a fun tower defence experience, I liked being able to shoot arrows, and the fact that you could infuse the arrows with an element was cool, altho for me it sometimes did not work, I think during 2 of the waves I could not infuse any of my arrows. I liked the graphical style, it looked pretty nice.

What aspects of the game did you not like?

I did not like the fact that you don't really know which spawn point the enemies will come from each wave, I like to plan out where to build before a wave starts. I kind of wish we start with some money so that we can a starting tower. I kind of prefer TD games where towers matter more than me, if you know what I mean? Like, I want the towers to do more dmg than me so that the towers matter more, I felt like i did 70% of the dmg with my arrows while the towers did 30%, I prefer it to be the other way, but thats just me, I know a lot of people prefer the way this game has it.

Herballistic

How challenging is the game? If it is too difficult or too easy please suggest why.

First time through I'd have said yes, but only in that you can't start with a tower, the bow seems underwhelming, and dodging the purple blasts isn't always effective in avoiding damage. After playing a bit more, nah, it's great difficulty-wise. One tiny tweak would make it excellent, but I'll get to that later. The map layout is procedurally generated. Does the generated path provide for an interesting experience? Yeah, it seemed handcrafted to drag my attention every which way during combat, but in a good way. Each tower got good range, fair overlap in fire zones, it was pretty well done for RNG. I'd love to see a couple more areas, or possibly a handful of pre-gen'd areas that it randomly picks from to start on. No same path through the levels on each run due to shuffling the levels.

Are the controls intuitive? Please suggest any possible improvements.

Grip to menu was odd, but it worked. I'd suggest using the menu button on the Vive for the menu, but after about 5 seconds grip for menu just clicked and it worked out fine. More of a personal taste thing, really. Otherwise, it's straightforward bow use and menu pointing, no point reinventing the wheel here.

Do you feel like the game rewards you for progressing through the waves of enemies?

Mostly, yes, but with one caveat. Defeating the dragon and getting all that bank, but not being able to use it makes me sad.

Are the graphical representations of game entities such as the towers, bridges and spirits easy to see and recognise?

Yeah, everything was really clear. Go here, shoot these, this is what a tower's Spirit is, these are the jerks shooting at you. All made sense.

Please suggest a feature that would make the game more fun and engaging.

Broad idea, make things upgradable, even just adding an extra spirit to a tower would be great. Like, first Spirit is the "main" one, second does less damage, but shoots independently and can combo well with the primary tower Spirit. Along the same line, upgrade(s) for the bow. More power, more stagger, a Spirit for the bow, anything like that would be 100% positive change.

What aspects of the game did you like?

Pretty much all of it. Honestly it's good enough as-is I'd like to use it during the VR demo I'm trying to put together for work. It's simple, fun, and is about as hard as you make it on yourself. With more levels and general expansion on your current ideas, this could be a great \$5 game, maybe more depending on scale and replay ability down the line. To sum up, my overall thoughts were that this is a great game for kids, folks new to VR, and anyone looking to just have a bit of quick, structured goofing off in VR. Being able to assist your towers manually is awesome, the pathing is good as is, you really can't beat The Lab's bow as it stands, and the premise is simple enough that anyone can get it.

What aspects of the game did you not like?

The arrows don't feel like they really do much past wave 3. Most importantly, staggering an enemy mage doesn't cancel or interrupt their attack. Killing them doesn't interrupt their attack. If you added working interrupts, and made the bow do a little something more on its own, e.g. slow enemies the more times you hit them with arrows sticking to them, stun just a bit longer, maybe more base power, then it'd be top shelf gameplay.

Fuzznutty

How challenging is the game? If it is too difficult or too easy please suggest why.

Not too difficult no, completed it on my first try, though I would say the enemies felt a bit too "bullet (or arrow) spongy". Felt like artificially big health bars and unless I'm wrong no crit shots?

The map layout is procedurally generated. Does the generated path provide for an interesting experience?

I liked the map layout, meant I had to kept dodging around to different towers to catch the twisting and turning enemies.

Are the controls intuitive? Please suggest any possible improvements.

Took me a long time to figure out how to start the game. I would personally move the menu to the menu button on the vive controllers rather than the grip buttons?

Do you feel like the game rewards you for progressing through the waves of enemies?

Very little rewards apart from the gold, didn't really understand which towers I should use either. Felt like a fair amount of gold each round though.

Are the graphical representations of game entities such as the towers, bridges and spirits easy to see and recognise?

Yep, great simplistic art style which I quite liked.

Please suggest a feature that would make the game more fun and engaging.

Critical shots would be a big plus from me (i.e. head shots etc.). Perhaps traps etc. as well as towers.

What aspects of the game did you like?

Great bow-and-arrow-play.

What aspects of the game did you not like?

Lack of sounds and general hard-to-figure-out menu system and goal, no real instructions within game itself.

Searangerx

How challenging is the game? If it is too difficult or too easy please suggest why.

I felt the game was a bit too challenging. The monsters seemed to soak up a lot of damage and the first round with no towers was very tedious. I felt the game was forcing me to sit as far back as possible to make use of the increased damage from range mechanic but the map layout didn't always make this possible and to be honest sniping things from far away wasn't particularly fun with how much missing and adjustment was involved.

The map layout is procedurally generated. Does the generated path provide for an interesting experience?

I found the map layout a bit confusing. It was never clear where the monsters were going to be coming from or where they were going.

Are the controls intuitive? Please suggest any possible improvements.

Everything with the bow felt fine. I enjoyed pointing my arrow into the spirit to super charge it. The menus were a little wonky. I think you'd be better suited to having the aiming laser appear with the menu rather than it being a separate button push after the menu is open.

Do you feel like the game rewards you for progressing through the waves of enemies?

It felt like the game escalated way too quickly by having multiple waves at the same time by only the second level and the tanky horse riders by the third.

Are the graphical representations of game entities such as the towers, bridges and spirits easy to see and recognise?

Everything is easy to tell what it is. No problems there except for what path the monsters will be taking. It would be nice to get a warning before the wave starts rather than having to teleport to the nearest tower and quickly build a defense as soon as the round starts.

Please suggest a feature that would make the game more fun and engaging.

I'd like to see damage counters in the game so I feel like I'm getting constant feedback about how hard my arrows and towers are hitting. The charged arrows costing money should be removed. It's the best mechanic in the game and having it use the same resource as your towers is a little weird. I'd rather have an unlimited supply with a recharging system. Either recharging the energy to use it over time or gaining the power back with kills. It would also enable more upgrade paths for the player to purchase outside of towers. If say the player started with 3 charges which recharge 1 every 10 seconds and could buy more charges with gold instead of towers.

What aspects of the game did you like?

I liked the bow and arrow mechanics and I especially enjoyed raising the arrow towards the spirit to charge it up. I like the design of the monsters and spirits they all fit nicely together. I think you've got a pretty solid core for a tower defense game.

What aspects of the game did you not like?

I found the monsters had a bit too much health especially early on when the player has very few towers. As I mentioned above I found the lack of foreknowledge about where the monsters were going to be mildly irksome.

rodwin0

How challenging is the game? If it is too difficult or too easy please suggest why.

It's not difficult, at all, except for the boss. Once you get enough upgrades, the creeps aren't a threat. The boss is a big one, as it shoots you big ball of fires you can barely dodge, except from moving to a different tower.

The map layout is procedurally generated. Does the generated path provide for an interesting experience?

Yes, it's working. However, the changes from one run to the other are not visible. You should give your path more designs, more details to make them recognisable.

Are the controls intuitive? Please suggest any possible improvements.

The arrow should be perpendicular with the controller. As it is now it's uncomfortable as your arrow wrist is twisted each time you want to shoot. The menu isn't very practical. The button to display it should be the "menu" button, set above the trackpad. It should also appear in front of the view. Setting it upper makes it difficult to see, use and can lead to some issues, like keeping it open while shooting arrows.

Do you feel like the game rewards you for progressing through the waves of enemies?

Many money is awarded, however, we don't know how it's given. The enemies are too much looking alike from one wave to the other.

Are the graphical representations of game entities such as the towers, bridges and spirits easy to see and recognise?

No. The towers should stand out more, specially when inhabited by a spirit. It's too white. I would recommend to change the tower colour depending on its spirit powers, or many to add a dome on top of it?

Please suggest a feature that would make the game more fun and engaging.

The spirits should be sellable, upgradable and switchable from one tower to another. The creeps should be different from one wave to another. If you make them quicker, or tougher, or magically protected, they should show it, by a glow, rays or a different skin colour. When an arrow hit a creature, it should display better the impact point, and award the player more money for his awkeye. The player should be able to regain life, using money, or by using a life spirit maybe?

What aspects of the game did you like?

It's simple to understand.

What aspects of the game did you not like?

The use of the colours : too white No difference between one enemy and the same one faster The animation. You should animate the continuously moving legs of the creeps separately from their upper body, so they doesn't seem to float while taking shots or shooting balls. The sound, not enough changes in the music and too few sound effects. When a creep is hit, it doesn't show any damage figure, no feedback except the always same hit animation. When a creep dies, it doesn't show the money awarded. During a wave, the player doesn't know the number of monsters to defeat, making it hard to know when a wave is finished or not. Nothing seems to happen when a monster reach the end of the path. I hope this will help.

m0jamb0

How challenging is the game? If it is too difficult or too easy please suggest why.

I don't think it is difficult per say, but I would lower the health of the enemies in the first few waves and maybe add a few to make it more fun. I found it frustrating having to hit the same guy 10 times to kill it.

The map layout is procedurally generated. Does the generated path provide for an interesting experience?

I liked the layout and the ability to move between towers is very cool.

Are the controls intuitive? Please suggest any possible improvements.

Outside of trying to figure out how to bring up the menu (I didn't read the manual on purpose) it was pretty intuitive. I would suggest maybe a small on screen blurb about the menu when you pick up the bow and arrow.

Do you feel like the game rewards you for progressing through the waves of enemies?

If you have the menu open, yes, because you can see the gold going up. I would suggest having a graphic over an enemy you just killed showing how much gold you just earned.

Are the graphical representations of game entities such as the towers, bridges and spirits easy to see and recognise?

Yes

Please suggest a feature that would make the game more fun and engaging.

Maybe allow players to place/move the towers between each wave if they want.

What aspects of the game did you like?

Being able to move to different towers. Overall I enjoyed it and look forward to seeing its progress.

What aspects of the game did you not like?

Strength on enemies is the early waves.

callMeSIX

How challenging is the game? If it is too difficult or too easy please suggest why.

Its seems hard at first but I think that's new environment new games and new features. After a few play throughs I knew where to place my towers. By my sixth attempt I was much more aware of my surroundings and was able to dodge all attacks until wave 6 (dragon wave).

The map layout is procedurally generated. Does the generated path provide for an interesting experience?

The map layout did make for a great experience, and gave great choices where to attack from, great mixture of high and low.

Are the controls intuitive? Please suggest any possible improvements.

The controls felt very good, no improvements or critiques. I did default to shooting from the hip.

Do you feel like the game rewards you for progressing through the waves of enemies?

Yes I do, good cash value, I think everyone likes a little character bonus ill discuss in response six.

Are the graphical representations of game entities such as the towers, bridges and spirits easy to see and recognise?

Mixed answer on this question, but I would say I have a few improvements. Bow, arrow, enemy movement, the town hall, and the back ground all look fantastic. I think ground texture of rocks or lava would add a lot, The path way should be more contrasting to the pillars but the glowing green when highlighted is great. For enemy's I think the attacking units should be a bit more contrasting as well, don't add colour or glowing, its a little over done in VR.

Please suggest a feature that would make the game more fun and engaging.

- A) Mixing tower powers - I only built exploding towers in my later tries, I didn't see much benefit in building the other 3, but if later levels you could add a second (exploding poison tower) that would be neat.
- B) Maybe unlock a more powerful bow with a damage 10% or 20% damage increase, its nice to level your character.
- C) Before the first wave have big arrows showing the spawning areas, the third was missed on attempt 1
- D) A few sounds would add a lot, Dragon roar when he shoots and a noise when the ranged guys shoot e)my arms were tired a cross bow or arrow turret would be cool but now I'm nagging.

What aspects of the game did you like?

- A) I liked the motion- at no time did I get sick or disoriented, which I am prone to.
- B) Smooth controls.
- C) Great mix of enemies - the horse guys are a real challenge.
- D) Different heights make a good feeling of space in VR.
- E)The tower concept is good.

What aspects of the game did you not like?

I wouldn't say I disliked anything, I outlined a few glitches and problems I had starting out but I think its fun. I don't know how much development you want to do, but different levels could be crazy different. Level 1 castle dragon, level 2 space and ships, level 3 street with bad guys and guns. Thank a lot, I had a good time.

space_goat_v1

How challenging is the game? If it is too difficult or too easy please suggest why.

I found it somewhat difficult. It felt like I had to spam arrows all the time and waves could easily out pace me and once it got to one wave on one end and another in the other end I couldn't keep up.

The map layout is procedurally generated. Does the generated path provide for an interesting experience?

It makes it random and fresh, but I think to be more interesting a variance in the aesthetic should be ideal. Like some wooden bridges, or random stair cases etc. It may be out of the scope of your project though.

Are the controls intuitive? Please suggest any possible improvements.

Yes, I would like the menu to toggle off tho, and maybe move the health and gold to one of the hands or a belt. I prefer less clutter. But otherwise Tele and the bow work perfect. But if you didn't put in the guide how to make elemental arrows I wouldn't have known to dip it in the orbs.

Do you feel like the game rewards you for progressing through the waves of enemies?

Yes somewhat, I think it could be more rewarding, unlocks for weapons like two/three arrows. Things that don't cost money but one time upgrades. Also the towers feel weak. In TD games I feel like the power comes from a bunch of clusters of towers spamming at an ever increasing power creep from the mobs. It feels more like an archery game with a tower addition.

Are the graphical representations of game entities such as the towers, bridges and spirits easy to see and recognise?

Yes the mobs are nice, scale is good.

Please suggest a feature that would make the game more fun and engaging.

Asynchronous mode. I love when a friend can participate. Maybe they could add towers or they could snipe in 2D on another tower, or drop bombs on enemies. More the merrier. There was another TD student project I played where the flat player was a robot guy that was basically an fps player that could run around the map and assist.

What aspects of the game did you like?

Arrows work well, mobs look good, the overall base of the game works well. Just could use more features.

What aspects of the game did you not like?

Other than my suggestions there was one glitch where the game over text overlaid the menu and I couldn't click on it. I had to altf4 and restart it.

Aredon86

How challenging is the game? If it is too difficult or too easy please suggest why.

I got to around level 5? Did try the game couple of times, so did not play it many times. The difficulty is ok right now, but the enemies need to give more feedback from being hit besides giving an animation of being hit by an arrow. Maybe a little stagger to their movement or even knock back depending the size of the enemy.

The map layout is procedurally generated. Does the generated path provide for an interesting experience?

It is nice to have the map procedurally generated. Gives a bit replay value for the game, and it worked in my runs.

Are the controls intuitive? Please suggest any possible improvements.

The controls are fine as they are right now. Maybe give arrows more power depending on how long you keep aiming. I found myself just spamming arrows too much.

Do you feel like the game rewards you for progressing through the waves of enemies?

Not really. The game needs more tower types and upgrade trees for them. Maybe even give possibility to buy new tower spaces with gold, for more optimal placement of towers and buying of barricades on the way of the enemies. Maybe even some melee and ranged troops. Think of kingdom rush.

Are the graphical representations of game entities such as the towers, bridges and spirits easy to see and recognise?

Yes.

Please suggest a feature that would make the game more fun and engaging.

Like I previously said, the game needs more tower types and upgrade trees and the possibility of choosing where you want to place them. Also I want to buy new bows, and maybe one time use arrows for them.

What aspects of the game did you like? What aspects of the game did you not like?

The game has a good base right now, but it is very barebones. Some work with enemy types, tower types and such is needed, (I am betting that you know this already). Also ways of impacting more the movement of the enemies, (staggering, slowing, knock backing, freezing).

mharmless

How challenging is the game? If it is too difficult or too easy please suggest why.

Most of the difficulty came from a lack of cues about taking damage. My brother lost one match after getting shot in the back repeatedly by the wizard dudes while he was shooting other monsters. On later play through I kept the menu open so I could see my health, which let me know immediately that I was being shot in the back. I think it could be made harder if it also had auditory or visual cues when taking damage.

The map layout is procedurally generated. Does the generated path provide for an interesting experience?

The environments are pretty bland, which makes the paths get more focus than it might otherwise. They would be more interesting if there were some terrain variation or more turns besides 90 degrees. Later on we figured out we could bank shots off the edges of the path and make the arrows slide down the lane and hit a monster very consistently. The interaction between arrows and the lane geometry was interesting and expansion on that would be welcome. Banking shots was one twist you could never see in a non VR game.

Are the controls intuitive? Please suggest any possible improvements.

Yes. Everything was very obvious in terms of conduction actions, after reading the manual. Understanding that the spirits you bought go over your head would not have happened quickly without the manual.

Do you feel like the game rewards you for progressing through the waves of enemies?

Not really. There doesn't seem to be an upgrade mechanic, so once all the towers were full we just spent gold on arrows and that did not feel rewarding. The impact of an explosive arrow feels low for its cost, so almost punishing rather than rewarding.

Are the graphical representations of game entities such as the towers, bridges and spirits easy to see and recognise?

Yes. The colour coding on the spirits made it trivial to tell which was which, and the enemies were distinct enough.

Please suggest a feature that would make the game more fun and engaging.

As above, more varied paths and some graphical variation. More tower spots, spirit upgrades. Bow damage did not seem to vary with pull length or distance, rewarding the player with more damage for farther shots or greater pullback; something along those lines would be welcome. Being able to summon an arrow for a tower you have a type of but that is different from the tower you are at might be more interesting. Deploying ice away from the ice tower is a more interesting choice. Perhaps storing arrows created at one tower to use elsewhere in a quiver? Going to a tower, stashing 10 expensive arrows, then grabbing them off my back where they might matter more is an interesting choice.

What aspects of the game did you like?

I liked the arrow banking, the notion that paths are randomised to get more replay-ability. I liked the massive size of the dragon, really stood out visually. Forced me to leave my perch to dodge its attacks. I like the music track, and the arrow pull sound coming out of the controller was crazy, no idea how you did that.

What aspects of the game did you not like?

There is a distinct parallaxing effect when looking at the far and middle distances. Looking down from a tower and shooting, all good. Look up into the distance a bit and move my head, the bridges seem to move at different rates and induce nausea. The closer to the horizon the worse it is. No problems at all looking down.

Gold and health are only visible in an intrusive menu, but you need both pieces of information while playing. So we ended up leaving the menu up. Those two might be better off displayed on the main play area.

The monsters were oddly quiet, felt like there should have been sound effects coming off of most of them. Maybe the wizard dudes work as silent ones, they look kinda grim-reaperish, but the dragon at a minimum is weird being silent. An auditory clue that the dragon is about to do his attack would be great.

Appendix C

Game Manual

The aim of the game is to kill the monsters before they reach the glowing temple. Monsters can spawn on any bridge with a red flag. You can kill monsters using either your bow and arrow or by summoning a “spirit” above your tower to shoot the monsters with magic. Killing monsters will reward you with gold with which you can purchase spirits.

Menu

The menu can be toggled by pressing either of the controller’s grip buttons. The menu will overlay on top of the rest of the game. Hold down the controller’s touchpad to activate a laser pointer. Use this laser pointer to point at a menu option and release the touch pad if you want to activate the menu option you are pointing at. In the menu you can purchase one of 4 different types of spirits or you can trigger the next wave by pointing your laser at the crossed swords. On the top left you can see your current gold value which is 0 at the start of the game and on the top right you can see the number of lives you currently have. If this number reaches 0 you have lost the game.

Towers

There are several towers place around the map. You can teleport to any tower you can see using your laser pointer to point at the tower you want to teleport to. Each tower can hold one magical spirit that will automatically target and attack your enemies.

Spirits:

There are 4 types of spirits that will attack your enemies:

Fire: Projectiles do 20 damage points.

Frost: Projectiles do 5 damage points and slows the enemy by 50% for 5 seconds.

Poison: Projectiles apply debuff that removes 3% of the enemies total health per second for 5 seconds. Dragon boss is resilient to poison.

Explosive: Does area of effect damage by applying 50 damage points to all enemies in a 12 meter radius.

Bow & Arrow

The bow mechanics are the same as the bow from “The Lab”. Use the trigger button to pick up the bow and hold the trigger button to pull back the arrow. Shots from further away do more damage than short distance shots.

Additionally you can upgrade your arrows by reaching the arrow hand towards the spirit above you.

The arrow will no carry the same particle effect as the spirit and do all damage and debuffs done by the spirit in addition to the regular arrow damage.

Price of arrows: Fire: 20G - Ice: 30G - Poison: 30G - Explosive: 50G

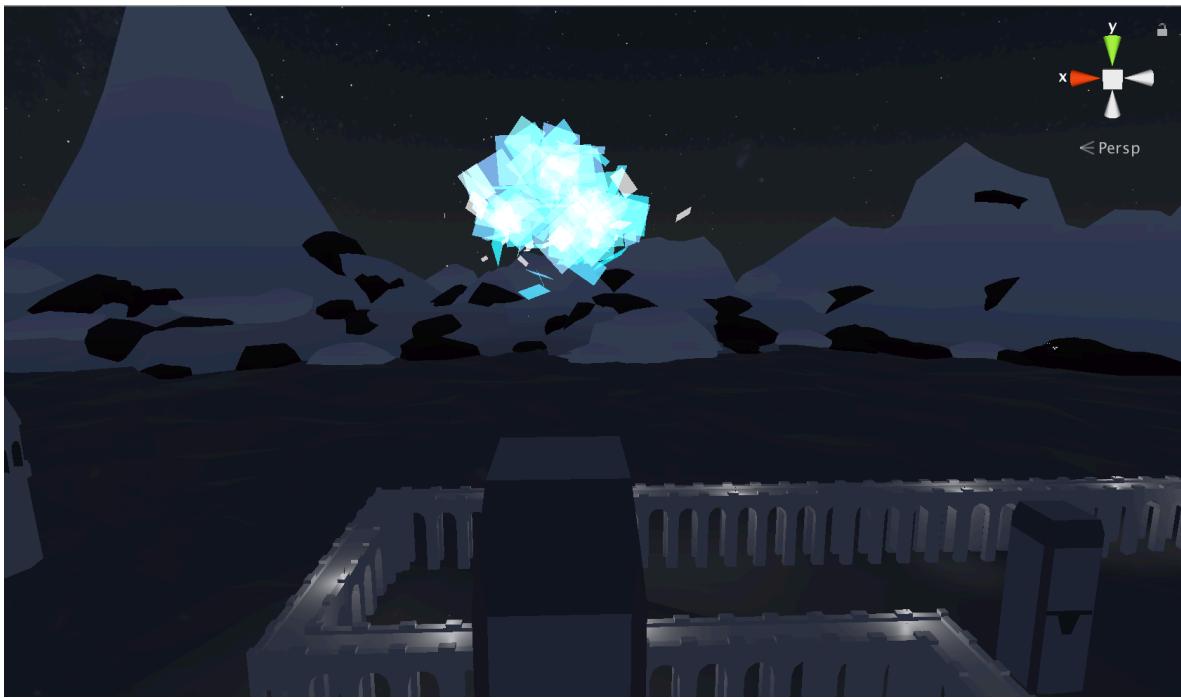
Be careful some monsters can shoot back at you and if they hit you, you will lose a life!



Menu



Tower



Ice Spirit



Fire Arrow

Appendix D

DOER - Virtual Reality Tower Defence Game

Background

In it's most basic terms a tower defence game is a game wherein a player builds defensive towers to prevent a group of enemies from crossing the game map. These towers shoot projectile at the enemy while they travel on a usually predefined path to stop them from reaching their goal location. Some argue that the genre dates back as far as 1990 with the release of the video game "*Rampart*" which involved protecting a castle from the projectiles of battleships in a nearby harbour¹. The player would have to build cannon towers to shoot at the ships and defend the castle. While this game may have been one of the ancestors in the evolution of the genre it is no longer what people typically think of as tower defence as it does not contain any enemy units trying to reach a destination.

The proliferation of this game type came in two waves. The first wave coincided with the dawn of adobe flash browser games. Tower defence games are not very computational intensive and lend them selves well to the web browser. The first flash tower defence game to gain wide spread success was "*Flash Element Tower Defense*" with nearly 500'000 users playing the game within one hour of release². The game implemented many of the genre defining features such as having enemies with weaknesses to different types of towers, requiring the player to think strategically as well as being able to upgrade towers giving the game a sense of progression. The second wave of popularity surged with advent of mobile gaming. Titles such as "*Kingdom Rush*", "*Plants vs. Zombies*", "*Field Runners*" and "*Bloons TD*" have seen tens of million of players toy with genre². Tower defence games have spread over a wide array of media due to their low computational demands, it is therefor a natural progression to execute this game type in virtual reality which in of itself is very computationally intense.

Description

The aim of this project is to design and implement a virtual reality tower defence game for the HTC Vive. The game will be set in a medieval fantasy world where the defenders are human units placed on towers. The oncoming enemy will be a variety of monsters such a goblin, trolls etc. The enemies will travel along a predefined path in order to kill the human king. The players can place different types of towers along that path. Every type of tower has different upgrade possibilities. Some towers are more effective against a certain type of monster than others.

The player will have the option to play the game from two perspectives. They can choose to play in a tactical mode where they experience a birds eye view of the map containing the towers. In this mode they can manage the game in a similar way to existing tower defence games by building new towers and upgrading existing ones to prevent the enemy from crossing the map. In the second mode the player can physically partake in the combat by teleporting into any of the towers. Each type of tower will grant the player a different set of weapons or powers. For example in one tower the player may be able to throw grenades at the enemy and in another fire a ballista at the oncoming horde.

The game will be implemented in Unity as it is the game engine with the best support for the HTC Vive hardware. The assets and world terrain will be sourced from the Unity asset store. If time allows, multiple game maps will be created.

Primary Objectives

- Implement a basic tower defence game in a 3D virtual space.
- Implement a tactical bird's eye view of the game where the player can strategically place towers and upgrade existing towers.
- Implement a first person view of the game where a player can engage in ranged combat from within one of the towers.

Secondary Objectives

- Implement challenging AI used by the enemy to cross the map.
- Implement a variety of interesting weapons for the player to fight with
- Implement a reward system where players can earn rewards that persist through game rounds.

Tertiary Objectives

- Create multiple maps/levels
- Create boss fights to make the game more diverse and exciting.

Ethical Considerations

The ethical considerations to make with this project are the same as with any other virtual reality video game. A person who is struggling coping with reality may be tempted to spend unhealthy amounts of time immersing themselves in virtual reality video games, such as this one. Furthermore, because the HTC Vive implements motion tracking, there exists a risk of users hurting themselves by hitting real life objects that are not present in the virtual world.

Resources

- Unity - Available for free online
- Fast GPU - Self supplied
- HTC Vive - Self supplied
- Testers - Friends



References

1. "Tower Defense – A Brief History." MyGaming. N.p., n.d. Web. 14 Feb. 2017.
2. "Computational Intelligence in Tower Defence Games 2014." IEEE Transactions on Computational Intelligence and AI in Games 5.3 (2013): 292. Web.

130012728