University of Plymouth

School of Engineering,

Computing, and Mathematics

PRCO304

Final Stage Computing Project

2019/2020

# Necromancer VR

Jack Fletcher

## Acknowledgements

I'd like to thank Dr Dan Livingstone for his support during this project as project supervisor.

Additionally, I'm grateful for the support of the ISS team, for guiding and supporting me throughout my degree.

## Abstract

This report details the programming processes followed throughout the development of a virtual reality (VR) based tower defence game. Through utilising and analysing relevant game theory, academic writings, and community feedback this report details the key features required for programming within VR and the preferable methods for reducing common VR related issues i.e. simulation sickness.

Additionally, the method of approach taken towards achieving the set minimum viable product (M.V.P) and minimum awesome product (M.A.P) is detailed. Including chosen methods and methodologies, software choices such as Unity, C# and SteamVR. Finally within this section project management tools are utilised such as Trello, Git and GitHub. The report details how these programs were employed, why they were chosen and their levels of success.Legal, social, ethical, and professional issues are addressed and considered where necessary and any negotiations required throughout the beginnings and length of the project are detailed and explained within this report.The stages of development are reviewed within this report and detail what was achieved through each step of development, any issues that arose, how they were handled and why methods were chosen.  The end project report evaluates the M.V.P and M.A.P requirements and to what degree they were met. It also explores the project goal and any criticisms that can be made of the final project.The project post-mortem analyses the overall project and the approach taken. It also evaluates the technologies chosen at the conception of the project now that it is completed.

Finally, this report summarises all programming and development decisions made throughout this project and evaluates any changes that would be considered if this project were to be repeated. It also analyses how any failings within the process could be taken as learning opportunities. Lastly, what can be taken from this experience by the author into future projects.

WORD COUNT: 9947

Github Link: https://github.com/jack-fletcher/Necromancer-Game

# 1. Introduction

The purpose of this project is to expand the author's experience with emergent technologies and provide a rich development environment to show a culmination of his ability as a developer. This project aimed to show how easily the author could adaptto a technology which has not been fully established within the chosen industry and apply previous knowledge to the subject, while taking into account design choices that must be considered in virtual reality.

Virtual reality (VR) is an emerging technology used in many industries, specifically in game development, the usage of VR has increased by almost 50%, with 1.29% of all Steam users reporting that they own a VR device in March 2020 (Steam Hardware Survey, March 2020). Additionally, with roughly 30,000 overall games on Steam, only 16%, or 5,000 of these are VR games. Therefore for such a prolific emerging technology, there is a requirement for the development of a wider variety of games to further development of VR technologies, not only for game development, but for commercial applications as well.

For the project to be considered to have reached its minimum viable product (M.V.P) the project needed a full game loop where the player can dig at a grave to unearth a body or part of a body, which can then be used to create a minion. These minions should then be used in a tower-defence style game, where the goal is to get to the end of the village and eliminate a special unit. The player should encounter and be able to create several types of units; including a basic 'combat triangle' with weaknesses and strengths for each unit. The player should be able to interact with the villagers outside of the tower-defence minigame for hints and dialogue options. This should all occur within virtual reality, with support for at a minimum; the Oculus Rift S headset.

For the project to be considered to have reached its minimum awesome product (M.A.P) the project needs to have all aspects of the M.V.P, with the following:

- A basic AI that can detect the player through either sight or sound
- Procedurally generated villages or multiple hand-crafted maps
- Support for multiple virtual reality headsets
- Support for playing without a virtual reality headset
- Individual limbs have stats which are applied to the unit and saved, with the highest stat becoming the units type, as opposed to having set units with predetermined stats.

The rest of the report will cover the background of the project, a short literature review, the method of approach used to drive the project, any legal, social, ethical or professional issues that were considered, and how the project was managed. Furthermore, the project will be broken down into a set of stages, and the project will be evaluated.

## 1.1. Background, objectives & deliverables

### 1.1.1. Background

The project is a single-player virtual reality experience with fantasy and tower defence elements. The idea is to have an explorable open world, with several interactable elements that drive game progression. There will be a set of game stages the player has to go through, however, the player can take this progression as slow as they desire. The project comes from the author’s desire to work with emergent technologies outside of game development, where being able to apply previous knowledge to problem domains the author may have no previous experience with, or research effectively to find a viable solution, will both become vital skills.

### 1.1.2. Objectives

The game will be set in a medieval fantasy environment and have a main goal; which is to eliminate the leader of a nearby village. This will require the player to create a small army of units, which becomes a sub-task for the player. Each unit requires the player to find limbs from deceased members of the village which correspond to a possible unit the player can create. The player will be able to interact with various villagers to find information on nearby graves and can visibly see what type of units there are in the village, and from there, strategise on what type of unit will be most effective to create.

### 1.1.3. Deliverables

This project will deliver a demo project, containing at least one level, as a Unity application with zero bugs that cause the application to fail, and with minimal bugs that affect the user experience during runtime.

# 2. Literature review

## 2.1. Locomotion in Virtual Reality

### 2.1.1. Precursor

Within virtual reality, various types of movement have to be considered to allow a game to be accessible to as many potential customers as possible. This is discussed within the ethical

section of this report. However, functionally, several methods of movement within virtual reality were analysed here, through the use of key players.
Locomotion is a key part of the user experience and poor design can deter the player from interacting with your unique selling point. Within VR, this has a far greater impact. A poor locomotion system can result in users becoming nauseous, disorientated, or light headed. Due to this, it is imperative that locomotion is researched for an optimal solution before development starts.

### 2.1.2. Locomotion Techniques

#### 2.1.2.1. Teleportation

One of these solutions is to use teleportation, which utilizes controllers to point towards a location, which is typically visually shown using an arc to the player's intended target (Figure 1).

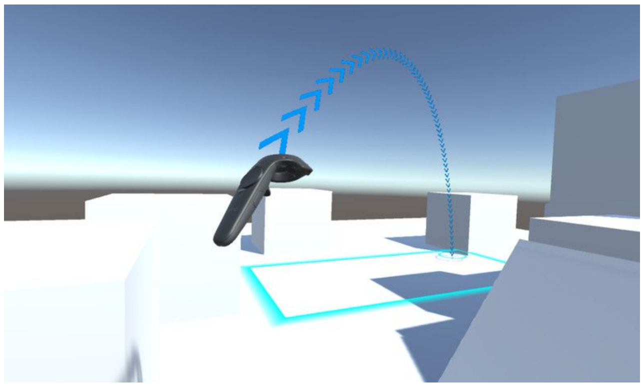


*Figure 1. An example of Teleportation within Unity*

This is typically used in games with limited movement, where the player's primarily using rotation to change their view of the environment such as those found in *Robo Recall* (Epic Games, 2017) as it is considered the least likely to cause simulation sickness (Christou, C. G., & Aristidou, P. (2017). Simulation sickness is a subset of motion sickness typically attributed to pilots who undergo simulation training for long periods of time (*Introduction to and Review of Simulator Sickness Research*, Johnson, 2005). Though is recently being attributed to sickness within virtual reality.
However, this has other adverse consequences. For example, depending on the implementation, the player may be limited to going between various predesignated teleportation spots, which results in predictable actions and may be easier to implement. However, this ruins immersion unless it can be implemented seamlessly into the story. On the other hand, the use of free teleportation may allow greater immersion but may result in the player entering unintended areas and requires greater user testing for game design. Additionally, teleporting in quick succession reduces the effectiveness of this method in minimizing simulation sickness.

### 2.1.2.2. Smooth Locomotion

Smooth locomotion is the method of manipulating a character by using at least one joystick on a controller, similar to a typical console experience. Typically used in open world RPGs such as Skyrim VR (Bethesda Softworks, 2017) and Fallout 4 VR (Bethesda Softworks, 2017), it becomes more immersive when moving around large areas due to better showing the scale of environments.

Smooth locomotion is the method most users will be familiar with when being introduced to VR, and due to this experience, will be intuitive. Additionally, this method would be preferable for someone who struggles with physical exertion, or a person who identifies as a person with a physical disability.

On the other hand, some users describe it as gliding around the world  (ul/lemonlemons, 2018) and immersion breaking due to having minimal physical action involved within the movement. Additionally, due to the variation between what their body is feeling and what their eyes are seeing, this method can be prone to motion sickness. This can be lessened by minimizing the amount of rotations the player has to make, or by rotating in stages, rather than by a continuous amount.

### 2.1.2.3. ArmSwinger

Arm swinging as a method of movement has been explored in both the academic and commercial sectors of VR.
*ArmSwinger* (ElectricNightOwl, 2017) is a VR locomotion library for the Unity Game Engine which allows the user to move within 3D space by manipulating your arms. Popularised by games such as *Creed: Rise to Glory* (Survios, 2018) and *Gorn* (Devolver Digital, 2018), it is touted as a ‘natural’ locomotion system as it mimics human movement when compared to other methods.

It is found that *ArmSwinger* induces minimal motion-sickness among users by tricking their brains into thinking they are moving around (u/TopinambourSansSel, 2018) and allows for use of the joystick for more precise movements, such as using held objects (u/rust_anton, 2018). Additionally, it was found that participants using arm swinging methods generally outperform a simple joystick in spatial orientation tasks (McCullough et al., 2015).

On the other hand, due to its inherent use of motion to generate velocity, it requires longer for the player to physically move their arms as opposed to simply tilting a joystick, which makes it impractical for covering long distance (u/DynaBeast, 2018). Additionally, it requires ‘[a]ctual

Physical Effort' (u/rtza, 2018) which can limit your user base to able-bodied players and can lower average play time for your user due to exhaustion. (In this context, able-bodied refers to literal bodies that are able, rather than exclusively physical or mental impairment). With games that have no serialization of data this can result in later stages of gameplay never being reached. It can also result in the player moving as an unintended consequence of moving their hand forward, for example in a typical action of thrusting with a sword.

### 2.1.3. Further research

An anonymous Questionnaire was given to students at the University of Plymouth concerning their experience with VR and their preferred type of locomotion. Of those that answered, 38.5% preferred teleportation as a method, suggesting this could be the optimal solution. However, 33% of people also attributed their motion sickness to teleportation, which paradoxically, also makes this less favourable.

### 2.1.4. Conclusion

With members of the community divided on a preferred type of locomotion, this debate can be seen as subjective. As a game developer, you want to reach as many customers as possible, which involves giving them a comfortable and immersive experience. To accommodate this, many games are giving player's the option to choose between various types of locomotion, which should be recommended while developing a new VR game. For Necromancer Game VR, a joystick movement system was implemented due to familiarity with the system, combined with 25% of people preferring it within the questionnaire, with plans to integrate multiple types of locomotion as a stretch goal, providing this does not negatively affect game balance.

## 2.2. Analysis of Key Players within Tower Defence

### 2.2.1. Precursor

The genre of tower defence is largely considered to have begun with *Rampart* by Atari Games (Ridley, 2013) and has evolved dramatically since then. This genre typically employs a variety of units which aim to reach a goal state within the game map. The goal of the player is usually to defend this goal state by placing obstructions or 'towers', in their various forms. These are analysed for their unique selling points (U.S.P.) to assist development of Necromancer Game VR.

### 2.2.2. Key Players

#### 2.2.2.1. Bloons TD 6

*Bloons TD 6* (Ninja Kiwi, 2016) is the latest installment in the well-known *Bloons (Ninja Kiwi, 2007)* franchine. Featuring various types of balloon as the enemy units, the player's goal is to use and upgrade a variety of towers to stop the balloons reaching the end of the map. This game presents a set of maps with a predetermined map, with much of the skill required being in systematically upgrading and replacing towers to more effectively eliminate the enemy. *Bloons* (Ninja Kiwi, 2007) utilises a ‘free mode’ style of gameplay, where the game consistently increases in difficulty until the player is defeated.

#### 2.2.2.2. Sanctum 2

*Sanctum 2* (Coffee Stain Studios, 2013) is a first-person tower defence game in a futuristic setting. It gives players a large open space, with the option to create their towers and obstructions in whatever shape they desire, with enemy units taking the best available path. Additionally, their U.S.P. is the use of player classes to assist their towers in a first-person shooter mode. This method bridges the gap between first person shooters and tower defence in an interesting and engaging way, allowing player's a greater depth of strategy within the same arena.

### 2.2.3. Further Research

Anti-tower defence is a subgenre of tower defence, where the roles are reversed. Players are typically limited by either budget or unit size and have to strategically choose the optimal units for a set map. In *Anomaly: Warzone Earth* (11 bit studios, 2011), the player instead uses special abilities to buff or assist their units within gameplay. This deviation from typical tower defence tropes was received well, with the game receiving a ‘Very Positive’ (Steam, 2020) review, implying that games can diverge from strict guidelines while still producing an enjoyable experience.

### 2.2.4. Conclusion

The variation between each case study within this review shows that having a unique selling point that does not conform to the typical tropes of tower defence, can still be a successful game.
*Necromancer Game VR* is supposed to bridge the gap between RPG games and tower defence games, by utilizing a subset of tower defence known colloquially as ‘anti tower defence’. In this game style, player's are moving towards a predetermined location and reacting to new information given by exploring. To utilize this data, enemies within Necromancer Game VR are predetermined and information is given to the player's through dialogue, allowing players to choose their units strategically before starting the tower defence gameplay, similar to how

towers are placed before the round begins. To combine this with RPG gameplay, enemies take the form of units, instead of towers, allowing further interaction between units.

# 3. Method of approach

## 3.1. Agile Methodologies

To deliver the project in an effective manner, Scrum was used to manage the project. Scrum (Schwaber & Sutherland, 1990s) is an agile framework that aims to iteratively develop a project. Scrum splits the project into sprints, based on their priority within the project deadline and aims to complete all tasks before the sprint deadline. Typically, agile methodologies such as Scrum are aimed at small teams, which results in small changes to the process to allow for use by an individual developer. Namely the removal of formal sprint meetings, such as the retrospective, review and planning meetings. Instead, those were informally considered when the sprint goal was outlined.

An alternative to this could have been Kanban (Ohno, 1988), which has continuous delivery of each feature, rather than splitting the project into sprints, each developer continuously takes tasks from the project backlog until completion.

Another alternative would have been Scrumban(Ladas, 2008) , which combines the methodologies set out in both Kanban and Scrum. Scrumban does not use sprint meetings, and has no requirement for role allocations, such as product owner and Scrum Master, which would have been preferable for a solo developer. However, contrary to Scrum's rigid sprint system, Scrumban uses a single continuous sprint until project completion which would not have worked efficiently for the developer.

## 3.2. Software

### 3.2.1. Unity Engine

Unity is a game engine created by Unity Technologies for use in developing both 2D and 3D games, used by both professional and indie developers. Due to this, its documentation and support via community forums provide excellent resources for rapid prototyping.
While engines such as the Unreal Engine can be more performant due to their use of C++, the author's minimal experience with the engine and the language would have resulted in this performance gain most likely being lost, or negligible.
Additionally, Unity does not require the developer to pay any royalties. However, it requires a professional license once your annual gross revenue exceeds $100,000, in comparison to Unreal Engine, which requires you to pay 5% royalties, with no costs to development.

### 3.2.2. C# (C Sharp)

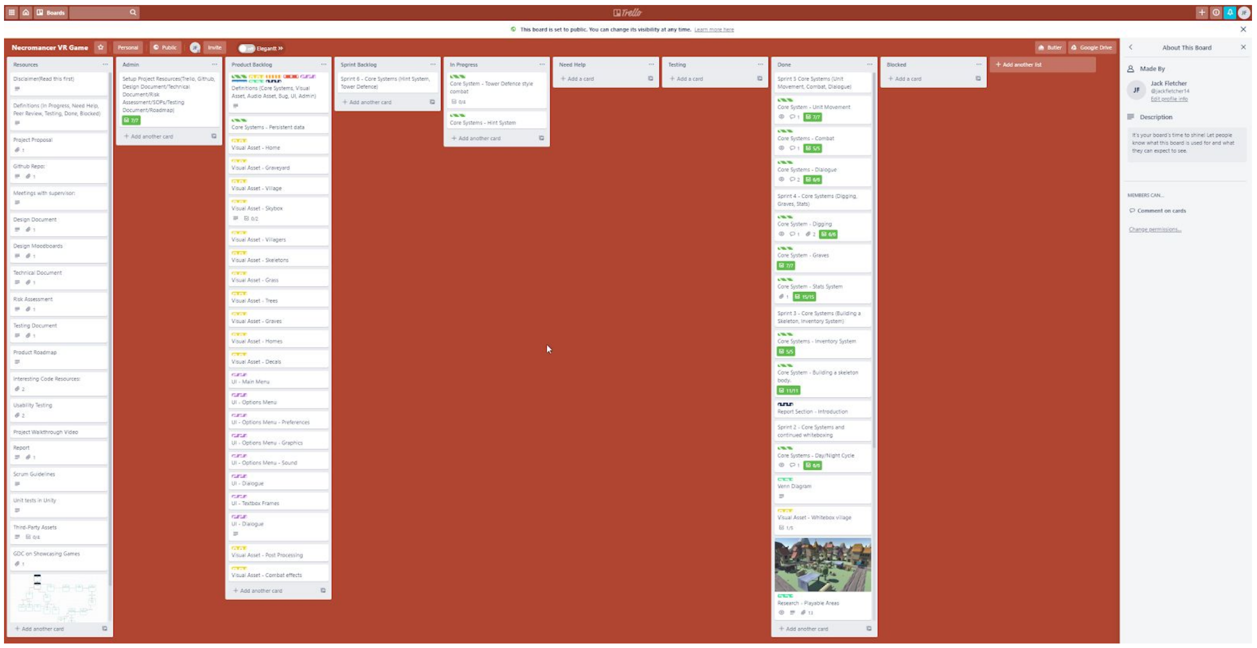
While within the Unity Engine both C# and JavaScript can be used, C# was the language of choice within this project. This was due to the author's previous experience with C# and the custom JavaScript variant used within Unity, known as UnityScript, having fundamental differences between itself and JavaScript, which makes researching common language issues problematic. Additionally, UnityScript was deprecated in Unity 2018.2, which would have resulted in having to downgrade on quality-of-life tools and improvements found over the last several iterations. Additionally, XML was chosen as the format to store all data in. This was chosen due to having XPath, a query language that allows you to select data from a document, and XML's extensible nature. Additionally, XML is also easier to read, and should this project be brought to market, the ability to pass game text to a third-party for localisation, without them needing access to development tools, would be crucial.

### 3.2.3. SteamVR

SteamVR was chosen for its compatibility with many headsets using its input manager, which only requires the developer to assign an action to each button. Additionally, the SteamVR documentation was more thorough than the Oculus Developer Guide.
One concern that has been raised is whether software using SteamVR can be published to the Oculus Store (Wekthor, 2018). bradweiers states 'If the OpenVR dll ends up being included in your build you will get rejected from Oculus Store submission with a message'. Due to SteamVR being a wrapper for OpenVR, it is likely this problem will persist, which may result in the application being unable to be published on the Oculus Store. However, when considering the twenty million users Steam has in March 2020 (Statista, 2020***)*** and how typically, users prefer to keep all their games in one launcher (u/Kamsides, 2019) it would be better to use the framework which allowed for access to a variety of different HMDs.

## 3.3. Project Management Tool

To visualise the Scrumban board, Trello (Atlassian, 2011) was used. Trello is a Kanban-style web application, suitable for organising data into a layout typical of a Scrum board (Figure 2). It was chosen due to experience with the application and its easy to use layout.


Necromancer VR Game

This board is set to public. You can change its visibility at any time. Learn more here

Resources
- Disclaimer(Read this first)
- Definitions (In Progress, Need Help, Peer Review, Testing, Done, Blocked)
- Project Proposal
- Github Repo:
- Meetings with supervisor:
- Design Document
- Design Moodboards
- Technical Document
- Risk Assessment
- Testing Document
- Product Roadmap
- Interesting Code Resources:
- Usability Testing
- Project Walkthrough Video
- Report
- Scrum Guidelines
- Unit tests in Unity
- Third-Party Assets
- GDC on Showcasing Games

Admin
- Setup Project Resources(Trello, Github, Design Document/Technical Document/Risk Assessment/SOPs/Testing Document/Roadmap) 7/7

Product Backlog
- Definitions (Core Systems, Visual Asset, Audio Asset, Bug, UI, Admin)
- Core Systems - Persistent data
- Visual Asset - Home
- Visual Asset - Graveyard
- Visual Asset - Village
- Visual Asset - Skybox 0/2
- Visual Asset - Villagers
- Visual Asset - Skeletons
- Visual Asset - Grass
- Visual Asset - Trees
- Visual Asset - Graves
- Visual Asset - Homes
- Visual Asset - Decals
- UI - Main Menu
- UI - Options Menu
- UI - Options Menu - Preferences
- UI - Options Menu - Graphics
- UI - Options Menu - Sound
- UI - Dialogue
- UI - Textbox Frames
- UI - Dialogue
- Visual Asset - Post Processing
- Visual Asset - Combat effects

Sprint Backlog
- Sprint 6 - Core Systems (Hint System, Tower Defence)

In Progress
- Core System - Tower Defence style combat 0/4
- Core Systems - Hint System

Need Help

Testing

Done
- Sprint 5 Core Systems (Unit Movement, Combat, Dialogue)
- Core System - Unit Movement 7/7
- Core Systems - Combat 5/5
- Core Systems - Dialogue 6/6
- Sprint 4 - Core Systems (Digging, Graves, Stats)
- Core System - Digging 6/6
- Core System - Graves 7/7
- Core System - Stats System 15/15
- Sprint 3 - Core Systems (Building a Skeleton, Inventory System)
- Core Systems - Inventory System 5/5
- Core System - Building a skeleton body. 11/11
- Report Section - Introduction
- Sprint 2 - Core Systems and continued whiteboxing
- Core Systems - Day/Night Cycle 6/6
- Venn Diagram
- Visual Asset - Whitebox village 1/5
- Research - Playable Areas

Blocked

About This Board

Made By: Jack Fletcher @jackfletcher14

Description: It's your board's time to shine! Let people know what this board is used for and what they can expect to see.

MEMBERS CAN... Comment on cards

Figure 2. The Trello board Layout. Also visible within Appendix A

## 3.4. Source Control

Git (Torvalds, 2005) is a distributed version control system (DVCS) typically used for software projects. Due to the author primarily working on a single computer, Git was used predominantly as a method to back-up the project and split the project into active and inactive branches, depending on what work was taking place. Git typically has extremely fast actions such as commits, due to the tool not needing to access a remote server for each commit, and only the hard drive. This also allows work to be completed and committed without an internet connection, without removing the ability to revert back to an individual commit.

In addition to this, the repository was stored on GitHub (Preston-Werner et al, 2008), a company that provides free version control hosting using Git, which can then be used in combination with GitHub Home, a GUI that allows for intuitive usage of Git without a command-line interface.

# 4. Legal, social, ethical and professional issues

As the majority of the assets, including code and UI elements, were created by the author specifically for this project, there is no concern with copyright infringement. However, due to time and ability constraints, third-party assets were used within the project, typically for visual assets.

## 4.1. Third-Party Assets

As third-party assets are being used, the assets that were chosen had to have several requirements. Firstly, for all visual assets, they all had to have a similar low-poly art style. To account for this, most assets were obtained through the same source, as typically, an artist will have a personal art style, and for the project to look professional, consistent visual assets were required. Therefore, Synty Studios was used for the majority of the visual assets, namely their '*Polygon Knights, Polygon Vikings, Polygon Adventure,* and *Simple Dungeons*' packs. These were obtained through a Humble Bundle (Rosen & Graham, 2010) , package, which was confirmed for use within commercial games, though the license is limited to a single seat. Furthermore, the use of SteamVR (Valve, 2014)  was used, as it provided a foundation for VR, with an easily expandable input system for multiple virtual reality headsets. If any code tutorials were used, a link to the tutorial is commented within an XML comment, at the top of the class. To make the project as commercially viable as possible, only assets with licenses that allowed for commercial use were used within the project. Furthermore, as the Unity Engine was used to develop the project, if the project was to become a commercial success, or be given a grant, the author would have to upgrade to a Unity Pro subscription if revenue or funding exceeded $200,000.

## 4.2. Considerations for VR

With virtual reality being a core technology for the project, considerations had to be made for how the player would interact with the environment, including movement, a core reason for simulation sickness within VR. To account for this, the author created an anonymous questionnaire aimed at previous experiences within VR, with the hypothesis that people would prefer methods such as teleportation, which limits movement to snappy actions. In conclusion, more people preferred teleportation, however, more people attributed their experience of simulation sickness to teleportation, so the result was inconclusive. However, the author chose to use joystick movement, a typically preferred method of movement, with a M.A.P for being able to choose various movement types to account for multiple types of player. In addition to this, VR suffers from being constrained to a physical space; where players need to be conscious of their surroundings, so they do not get injured. To account for this, players should use either Steam Chaperone (Valve, 2017) or Oculus Guardian (Oculus, 2020) to ensure they set a valid play area and stay within the allotted area.

## 4.3. User Testing

Usability testing that was undertaken through questionnaires and demo sessions was executed using students at Plymouth University. Any data that was used, was taken using a Google Form, which converts all data into an easily readable spreadsheet, with no personal data taken. Additionally, all participants were required to give consent at the beginning of the questionnaire or demo session, with a further requirement for consent as to whether their results could be shared anonymously on social media.

## 4.4. Pegi Rating

The Pan European Game Information rating, or PEGI rating, is used to classify games into sections that are appropriate for players below the age given for that rating. Due to this project showing unrealistic violence towards human-like characters, I have given this project the unofficial PEGI 7 rating, with plans to get an official rating should this project become commercially viable.

# 5. Project management

## 5.1. Workflow

At the beginning of each sprint, work was moved from the project backlog to the sprint backlog, indicating it was ready to be worked on. Tasks were chosen based on the priority given to it previously, indicated by how high it is in the project backlog list. The higher the task, the more priority it was given. During development, it was then moved to in progress, and any further work or research that may have been done during the task, documented within its checklist, or in the comments below. After each user story within the checklist is completed, it is moved to testing, where its function is tested and documented within the testing document spreadsheet. Once each user story has successfully passed its tests, the task is pushed to GitHub (Preston-Werner et al, 2008), and merged to release, and a new task is taken from the sprint backlog.

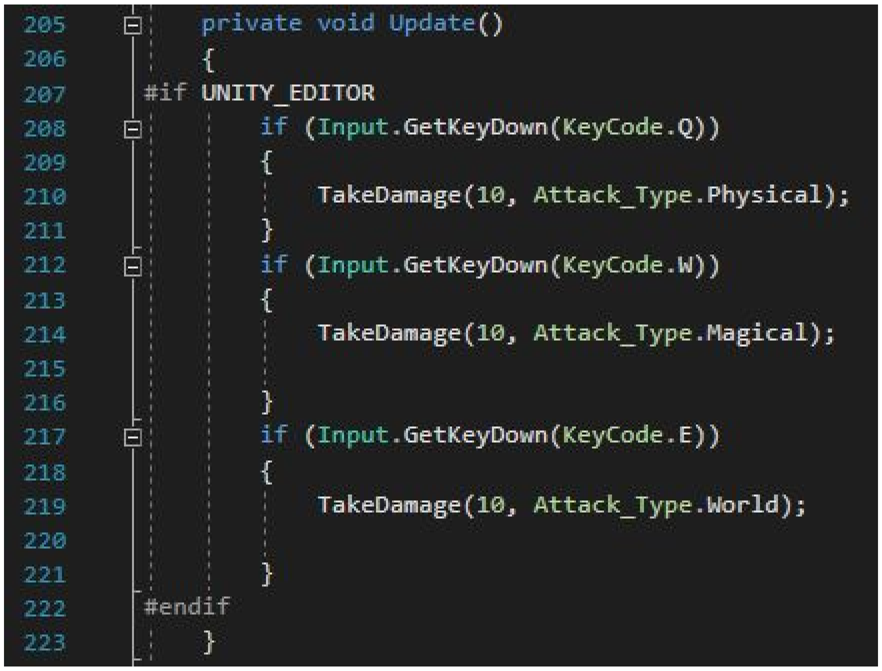
## 5.2. GitHub

Git (Torvalds, 2005), using GitHub (Preston-Werner et al, 2008), was used as version control within this project, and was split into several branches. The project used GitHub Flow, a lightweight workflow method which revolves around having several points for testing. The master branch was used as a production branch, aiming to be the most stable version of a release that could be used, should a theoretical stakeholder wish to see it, and was updated at the end of each sprint. The release branch had all work from the current sprint that had been completed, then a branch for each task that required completion. However, a main aspect of GitHub Flow is the pull request, which typically comes with code review. Working as a solo developer, I was unable to have a person or team review my changes, which negates a fundamental part of GitHub Flow. However, I was able to adjust this, and use this point to check my code for any code convention errors.

## 5.3. Testing

### 5.3.1. Testing Document

Within each sprint Scrumban requires each task to be tested before a merge to release, which I show within this testing spreadsheet. This document is split into sections, the most important of which showed the test that was performed, the expected result for this test, whether it was as expected, and any further comments. This was inspired by unit testing and further expanded on by writing preprocessor directives within testable scripts update loops (Figure 3).



```
    private void Update()
    {
#if UNITY_EDITOR
        if (Input.GetKeyDown(KeyCode.Q))
        {
            TakeDamage(10, Attack_Type.Physical);
        }
        if (Input.GetKeyDown(KeyCode.W))
        {
            TakeDamage(10, Attack_Type.Magical);

        }
        if (Input.GetKeyDown(KeyCode.E))
        {
            TakeDamage(10, Attack_Type.World);

        }
#endif
    }
```

*Figure 3. An example of functional testing within gameplay*

In the given example, the directive only occurs when the player is within the Unity editor, which ensures that should this not be commented out before being released, it would not affect consumer gameplay. The example tests whether the player correctly takes damage, and is expanded by a debug line within the TakeDamage method, which shows the output of the method.

Alternatively, Continuous Integration/Continuous Deployment(CI/CD) testing through the use of tools such as Jenkins (Kawaguchi, 2011) could have been a viable way to unit test the project. However, with the author’s minimal understanding of how CI/CD works within the Unity Engine, it was considered too high risk to learn within the short, three-month timespan of the project.

#### 5.3.2. User Testing

Due to the current pandemic of the COVID-19 virus and hygiene concerns with head-mounted displays, user testing was limited. I was only able to test with people within my own household and people who owned virtual reality headsets, which, according to Steams 2020 hardware survey (Steam, 2020) , only make up 1% of overall users on the Steam platform. These tests were unable to be monitored in person. As a result, a questionnaire was provided to those who tested the project. These were then gathered into anonymous Google spreadsheets to be analysed for data.

## 5.4. Conventions

To keep the project consistent, a set of conventions had to be established to make formal documentation easier. These include the use of code conventions, such as prefixes for local and global variables, XML comments, and the use of CamelCase for variable declaration and Pascalcase for method declaration. Additionally, a project structure was set to keep all files organised. This is developed in further detail within the technical document.

# 6. Stages

The project was split into several stages, based on importance. Stage one, was essential technologies and project set up, which had to be implemented before the project could continue, and took place in sprint one. Stage two and three, was composed of all core features and took place within sprints beginning with sprint two, and ending with sprint seven. Stage four consisted of combining all features created within stages two and three, and adding all nonessential features such as health bars, polish and quality of life changes and took place in sprints eight, nine and ten.

## 6.1. Stage 1: Initial Setup

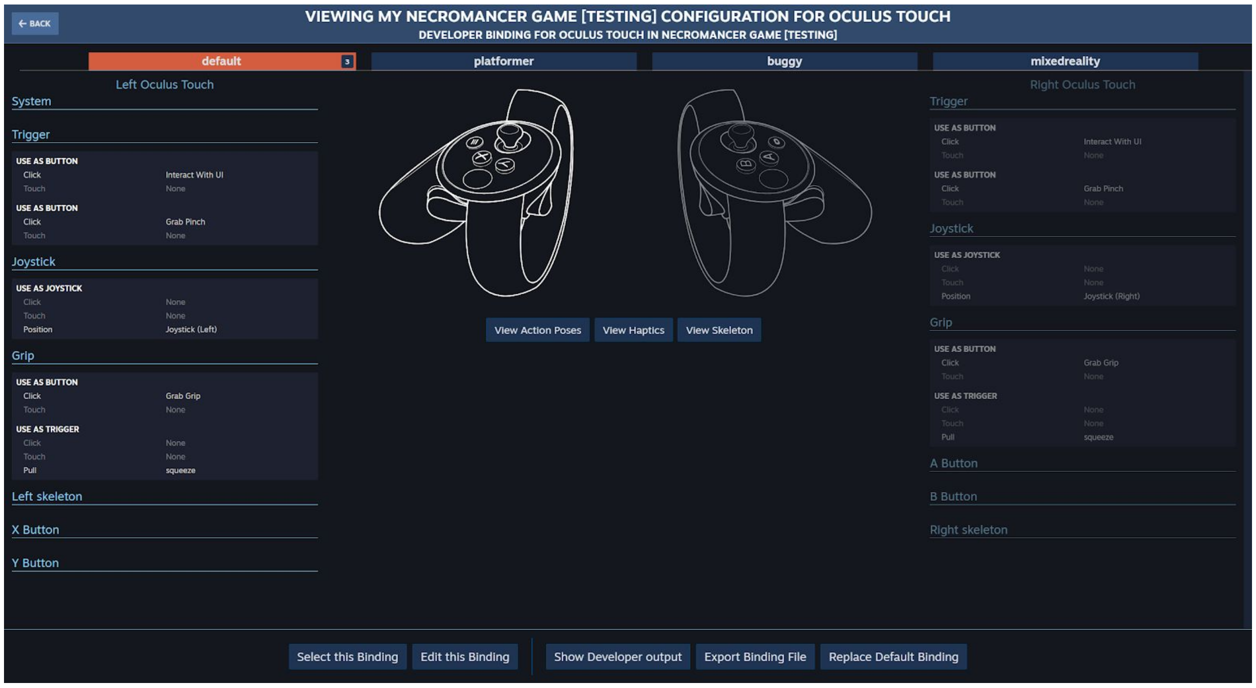
At the beginning of the project, project management tools were set up. This included setting up a Kanban board using Trello by creating a set of lists, the most important of which are the product backlog, in progress, testing and done sections. Functions within the project were then split into tasks, shown as cards, and further developed using lists when applicable to function as user stories.
In addition to this, a GitHub repository was set up. This was split into two branches and left private for intellectual property security. To maintain consistency throughout the project, a design document with a high-level overview of how features should behave and look was created, alongside a technical document for a low-level description, including coding conventions and project structure. Additionally, a risk assessment was created, detailing procedures that should be undertaken to continue the project, should any issues arise.

Finally, the testing document was created.
With this completed, the Unity project was set up and initialised on GitHub with the file structure set out in the technical document. A test scene was then created and used as a white box scene, where features could be tested without impacting the main scene and to test the scale of objects created during the project.
Initially, the Oculus Integration package was used to implement virtual reality within Unity. However, this had significantly worse documentation than the SteamVR package, and incidentally, only allowed for development for Oculus devices. In comparison, Steam allows you to set up an input system for various virtual reality devices, so it was a far better choice for developing a game that could be accessed by a variety of devices.

## 6.2. Stage 2: Core Systems - VR

### 6.2.1. Movement

SteamVR has a custom input handler for interacting with virtual reality headsets, which takes a group of actions, assigns a type to them, and allows the developer to set a default and suggested set of controls, similar to standard input systems. Where this differs, is that this can be applied or edited for any number of virtual reality headsets, which extends its use indefinitely in the constantly expanding technology that is virtual reality.
To accommodate for this, the M.V.P only required the project to have controls for the Oculus Rift S due to available hardware limitations (Figure 4).



VIEWING MY NECROMANCER GAME [TESTING] CONFIGURATION FOR OCULUS TOUCH
DEVELOPER BINDING FOR OCULUS TOUCH IN NECROMANCER GAME [TESTING]

Left Oculus Touch

System

Trigger

| | |
|---|---|
| USE AS BUTTON | |
| Click | Interact With UI |
| Touch | None |
| USE AS BUTTON | |
| Click | Grab Pinch |
| Touch | None |

Joystick

| | |
|---|---|
| USE AS JOYSTICK | |
| Click | None |
| Touch | None |
| Position | Joystick (Left) |

Grip

| | |
|---|---|
| USE AS BUTTON | |
| Click | Grab Grip |
| Touch | None |
| USE AS TRIGGER | |
| Click | None |
| Touch | None |
| Pull | squeeze |

Left skeleton

X Button

Y Button

Right Oculus Touch

Trigger

| | |
|---|---|
| USE AS BUTTON | |
| Click | Interact With UI |
| Touch | None |
| USE AS BUTTON | |
| Click | Grab Pinch |
| Touch | None |

Joystick

| | |
|---|---|
| USE AS JOYSTICK | |
| Click | None |
| Touch | None |
| Position | Joystick (Right) |

Grip

| | |
|---|---|
| USE AS BUTTON | |
| Click | Grab Grip |
| Touch | None |
| USE AS TRIGGER | |
| Click | None |
| Touch | None |
| Pull | squeeze |

A Button

B Button

Right skeleton

.

*Figure 4. SteamVR's input system (Valve, 2019)*

For movement, the JoystickLeft and JoystickRight inputs were set as Vector2's, which allowed them to be referenced in code using the SteamVR_Action_Vector2 class
The left joystick allows for movement, which is normalized and multiplied by the project's gravity settings and time.deltatime. This allows for movement to be consistent, due to not being frame-dependent. This is then used as a parameter for Unity's built-in Character Controller component to move the character.
The right joystick rotates the player by accessing the player instance's transform and rotating it by the value within the input vector.

### 6.2.2. Interactions

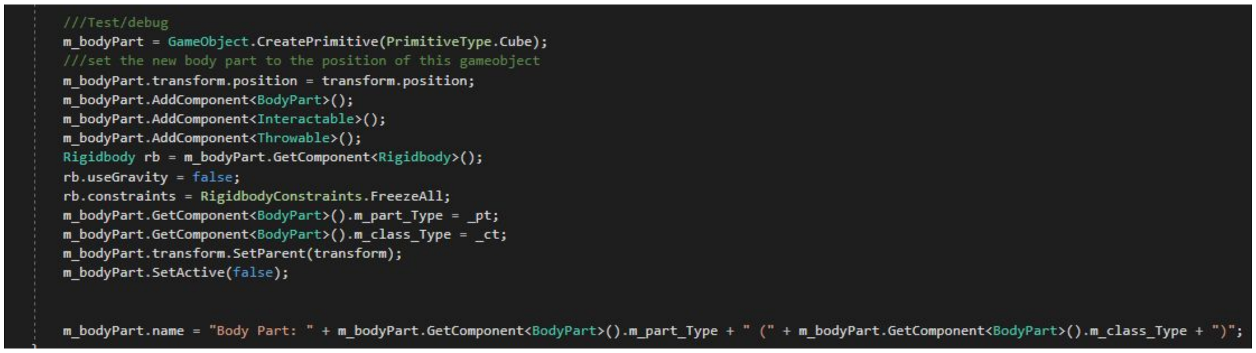
Base interactions, such as wielding and throwing within virtual reality were entirely handled by SteamVR's integrated scripts, known as 'Interactable' and 'throwable'. This was extended using a weapon class, which allowed for objects that had these components to be used as weapons within the game. This allowed for specific weapons to be used as single use throwables such as apples or buckets, more conventional weapons such as swords or axes, or grave-specific weapons such as pickaxes.

## 6.3. Stage 3: Core Systems - Game-Specific

### 6.3.1. Graveyard System

The Graveyard System is split into three sections. Firstly, there is a GraveManager script that keeps track of every grave within the scene and performs setup functions on each of them. It makes an array of each GameObject within the scene that has the tag 'Grave' then iterates through each GameObject, selecting its Grave script and Gravestone script that make up the two other sections of this system and calling their SetCanvasText and SetBodyPart methods. It then creates a random index, then selects the body part from Part_Type at this index. This is then repeated for the Class_Type. The Gravestone script is only used to set the canvas text of the gravestone, through the GraveManager.

The Grave script creates each body part through SetBodyPart. This method creates a body part, then sets up each of its required components. Additionally, this script is responsible for the grave's health, which is removed when the player hits it with a weapon. When the grave's health is reduced to zero, the grave is disabled and the body part is revealed (Figure 5).



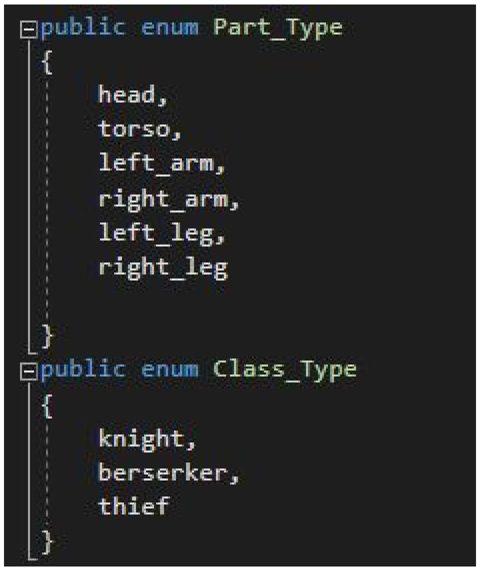
```
///Test/debug
m_bodyPart = GameObject.CreatePrimitive(PrimitiveType.Cube);
///set the new body part to the position of this gameobject
m_bodyPart.transform.position = transform.position;
m_bodyPart.AddComponent<BodyPart>();
m_bodyPart.AddComponent<Interactable>();
m_bodyPart.AddComponent<Throwable>();
Rigidbody rb = m_bodyPart.GetComponent<Rigidbody>();
rb.useGravity = false;
rb.constraints = RigidbodyConstraints.FreezeAll;
m_bodyPart.GetComponent<BodyPart>().m_part_Type = _pt;
m_bodyPart.GetComponent<BodyPart>().m_class_Type = _ct;
m_bodyPart.transform.SetParent(transform);
m_bodyPart.SetActive(false);


m_bodyPart.name = "Body Part: " + m_bodyPart.GetComponent<BodyPart>().m_part_Type + " (" + m_bodyPart.GetComponent<BodyPart>().m_class_Type + ")";
```

*Figure 5. The Grave Manager script*

A guard was set up to patrol the area to look for the player. To simulate vision, this guard follows a set path within the environment while casting a physics sphere. If a collider within this physics sphere is a part of the Player layer, it shoots a raycast from the guards eyes to the center point of the Player GameObject. If this raycast successfully reaches the player, it is assumed that there is a clear line of sight between the guard and the player and therefore the player has been spotted, which changes its goal state to the player's location. The guard keeps the player as its goal state until the player has left the area of the physics sphere.

### 6.3.2. Character Creation

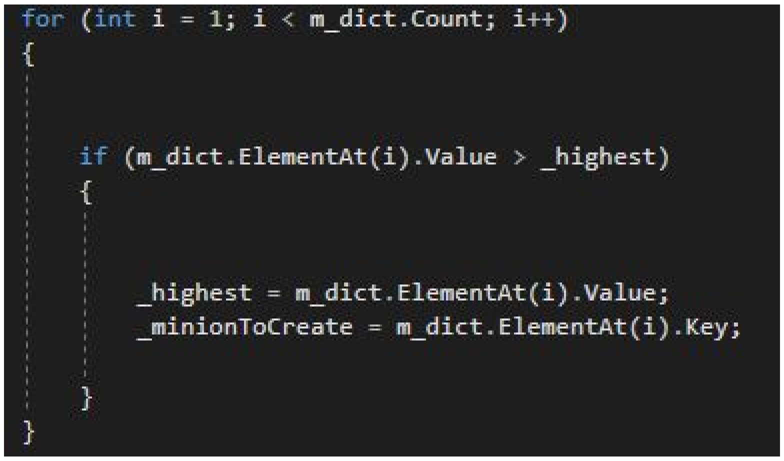
Character creation can be split into several sections. Firstly, each body part has a ‘Body Part’ script, which only holds a reference to two public enums, the Part_Type of the body part, and the Class_Type. These enums reference which body part the object is and which class type it derives from (Figure 6).



```
public enum Part_Type
{
    head,
    torso,
    left_arm,
    right_arm,
    left_leg,
    right_leg

}
public enum Class_Type
{
    knight,
    berserker,
    thief
}
```

*Figure 6. The definition of two enums within the enum.cs file*

Secondly, a GameObject within the scene acts as a placement guideline for the player. This GameObject then has several empty game object children which act as snap points at various points across the bodies. When its box collider is triggered, the SnapPoints script’s OnTriggerEnter method is called, which checks first if the GameObject that triggered OnTriggerEnter contains a BodyPart script, and if this returns true, checks what kind of body part it is. This other GameObject then has its interactable script disabled, it is removed from the player's hand, moved to the correct snap point and set as a child of that GameObject.
Within the CharacterCreator script, during awake, the dictionary m_dict is initialized with each value of the Enum Class_Type. This is used to count each instance of a specific class within a body. CheckForParts iterates through the snap points and if each element has a child, it adds to the counter set up in m_dict. After the iteration, if all elements had children, it creates a unit with the class set to the class type with the most counts within m_dict (Figure 7).



```
for (int i = 1; i < m_dict.Count; i++)
{


    if (m_dict.ElementAt(i).Value > _highest)
    {


        _highest = m_dict.ElementAt(i).Value;
        _minionToCreate = m_dict.ElementAt(i).Key;

    }
}
```

*Figure 7. The code that defined which minion to create*

## 6.3.3. Navigation System

This was set up using a singleton Navigation Manager which every unit accessed to see the path to be taken. The path was made up of empty GameObjects that served as nodes, which units could move between, until the final node. Originally, this was sorted by distance to the start point for an efficient path. However, this resulted in paths that may be undesirable, due to nodes potentially being closer to the start point later in the sequence, however not intended to be accessed first. The navigation itself is handled by each individual unit. Additionally, using Unity’s Gizmo system, each node is visualized using a blue line for a path and a red wireframe sphere for the point, for easier customization (Figure 8).

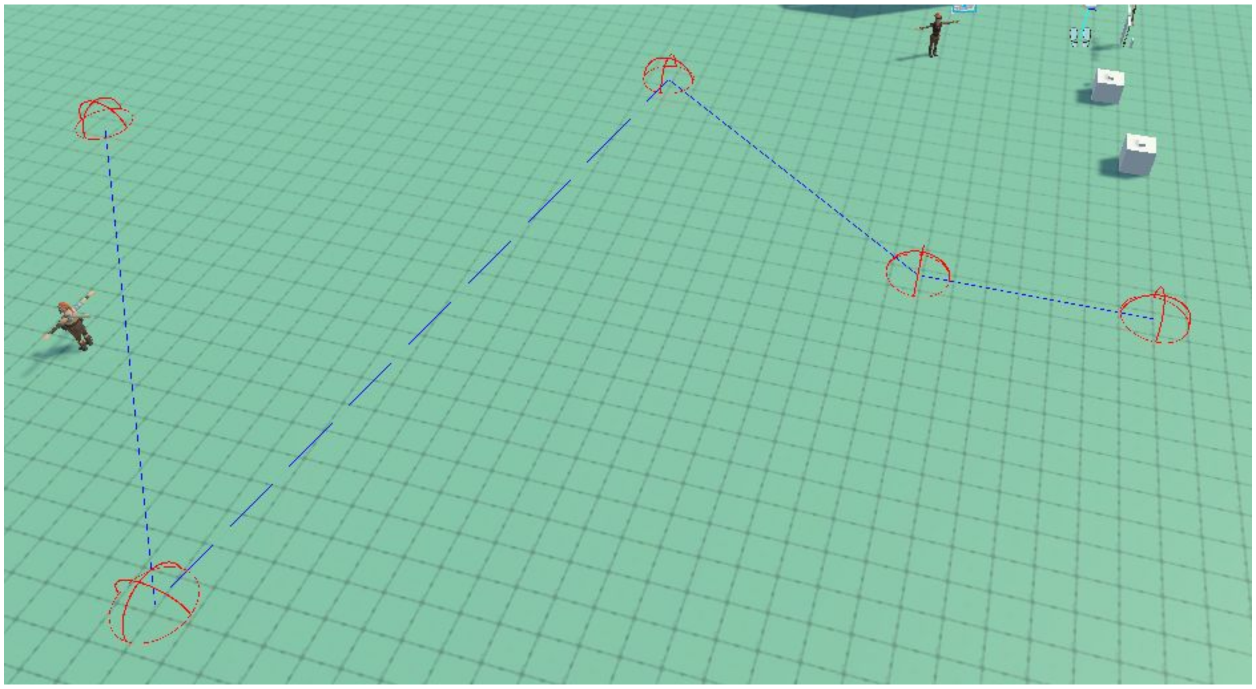


Figure 8. An example of the navigation system, within a test scene.

### 6.3.4. Movement

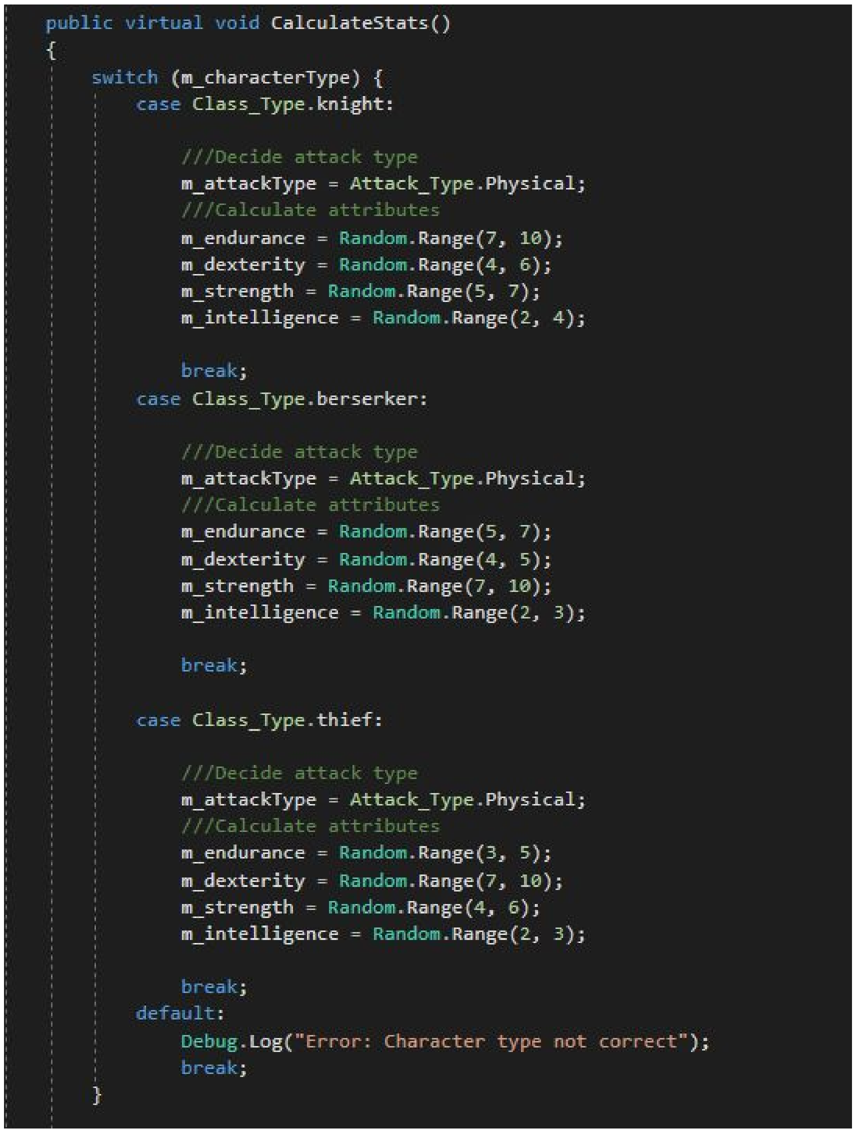
Movement was created using a goal system, the unit attempts to move towards a goal state, typically the next node of the navigation system, until it completes the path. For friendly units, this is controlled by the FriendlyController script and occurs within the SimpleMove method. This takes in a goalIndex, typically the next node of the system, and moves towards it using the CharacterController component. At this point, the unit checks for any points of interest such as enemies, and if so, moves towards them. After this, the unit continues along the predetermined path. In addition to this, enemy controllers are typically stationary, however, can use this same movement system, or patrol between two separate points.

### 6.3.5. Dialogue

The dialogue system is largely derived from a Brackeys tutorial *How to make a Dialogue System in Unity* (Thirslund, 2017), which creates three scripts, a DialogueManager, Dialogue, and a DialogueTrigger. This was expanded to show the sentence canvas based on proximity using VR. Additionally, this interfaces with the XML Manager to dynamically create dialogue based on the XML.

### 6.3.6. Character Stats

The character stats are typical of an RPG-Style game, with several attributes that have stats derived from them. Within the CharacterStats script, each attribute is derived randomly from a range based on its class type in the CalculateStats method. Additionally, the stats are derived from these attributes (Figure 9).

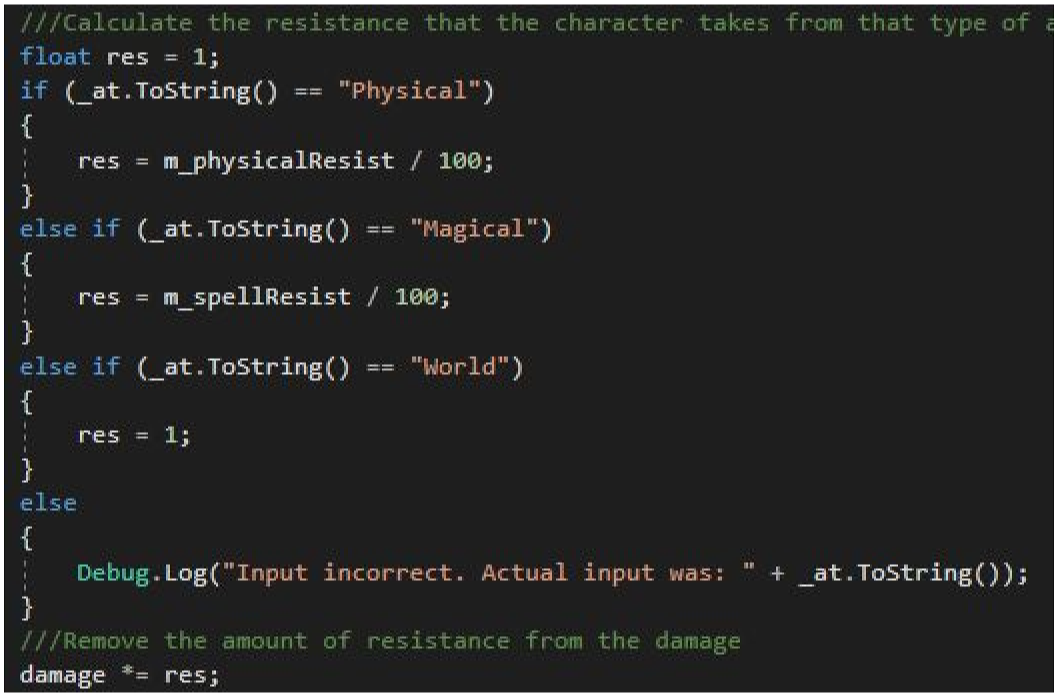


```
public virtual void CalculateStats()
{
    switch (m_characterType) {
        case Class_Type.knight:

            ///Decide attack type
            m_attackType = Attack_Type.Physical;
            ///Calculate attributes
            m_endurance = Random.Range(7, 10);
            m_dexterity = Random.Range(4, 6);
            m_strength = Random.Range(5, 7);
            m_intelligence = Random.Range(2, 4);

            break;
        case Class_Type.berserker:

            ///Decide attack type
            m_attackType = Attack_Type.Physical;
            ///Calculate attributes
            m_endurance = Random.Range(5, 7);
            m_dexterity = Random.Range(4, 5);
            m_strength = Random.Range(7, 10);
            m_intelligence = Random.Range(2, 3);

            break;

        case Class_Type.thief:

            ///Decide attack type
            m_attackType = Attack_Type.Physical;
            ///Calculate attributes
            m_endurance = Random.Range(3, 5);
            m_dexterity = Random.Range(7, 10);
            m_strength = Random.Range(4, 6);
            m_intelligence = Random.Range(2, 3);

            break;
        default:
            Debug.Log("Error: Character type not correct");
            break;
    }
```

*Figure 9. The code defining stat ranges for characters.*

### 6.3.7. Combat

Combat is also handled within the CharacterStats script and takes into consideration the attack type that was previously set. When a character is within range of an enemy, the Attack method is called. From here, a timer is created and compared to the attack speed of a unit, which has to be fulfilled before the unit can attack. When the unit attacks, there are three types of resistance, with each class having a resistance derived from their endurance or intelligence. These are used as a multiplier when calculating damage within the TakeDamage method, thus reducing the effectiveness of an attack. If the character is left with zero health, the character then dies and is set to inactive (Figure 10).

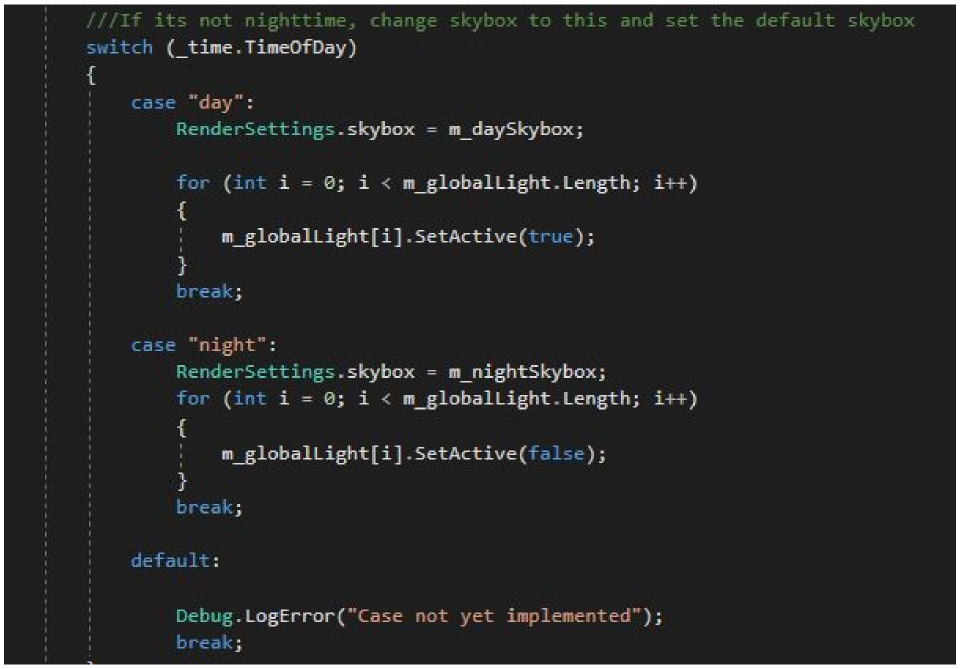


```
///Calculate the resistance that the character takes from that type of a
float res = 1;
if (_at.ToString() == "Physical")
{
    res = m_physicalResist / 100;
}
else if (_at.ToString() == "Magical")
{
    res = m_spellResist / 100;
}
else if (_at.ToString() == "World")
{
    res = 1;
}
else
{
    Debug.Log("Input incorrect. Actual input was: " + _at.ToString());
}
///Remove the amount of resistance from the damage
damage *= res;
```

*Figure 10. The code defining damage resistance.*

### 6.3.8. Day/Night Cycle

The time of day is set using a TimeManager, derived from Code First's observer pattern example *C# Observer Pattern Example* (NAME, 2019) which notifies each of its subscribers when the time of day is changed. To physically change the world, all lights that were to be turned off during the daytime were given the tag 'GlobalLighting' and assigned to an array. When the time of day changes, this is then turned off or on, and the skybox changes (Figure 11).

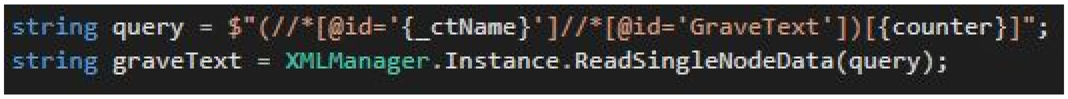


```
///If its not nighttime, change skybox to this and set the default skybox
switch (_time.TimeOfDay)
{
    case "day":
        RenderSettings.skybox = m_daySkybox;

        for (int i = 0; i < m_globalLight.Length; i++)
        {
            m_globalLight[i].SetActive(true);
        }
        break;

    case "night":
        RenderSettings.skybox = m_nightSkybox;
        for (int i = 0; i < m_globalLight.Length; i++)
        {
            m_globalLight[i].SetActive(false);
        }
        break;

    default:

        Debug.LogError("Case not yet implemented");
        break;
```

*Figure 11. Swapping lights based on time of day*

### 6.3.9. XML Manager

To allow for further translations to be made should the project become commercial, XML was chosen as the format to store game data such as names. Using System.Xml and System.IO, the XML file is searched for queries and returns the elements within (Figure 12), which can be used to fill grave names, or hints. For this, two methods were created - ReadChildNodeData and ReadSingleNodeData. These return either a specific node, or all children of the node that is found. This allowed the graves to be changed to use data from the XML file, making the grave system much more expandible.



```
string query = $"(//*[@id='{_ctName}']//*[@id='GraveText'])[{counter}]";
string graveText = XMLManager.Instance.ReadSingleNodeData(query);
```

*Figure 12. An example of a query that is sent to the XMLManager.*

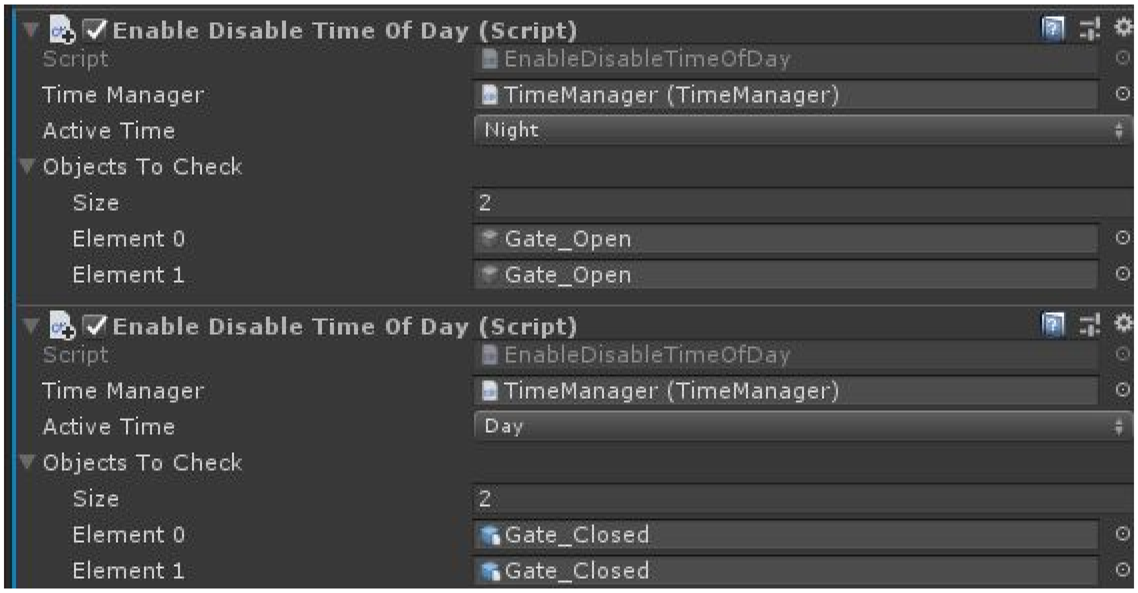
Additionally, this was used for the dialogue system where instead of typing the dialogue within the editor to set the sentence data, it was crafted within the XML file.

## 6.4. Stage 4: World Building

### 6.4.1. Visual

#### 6.4.1.1. Time Specific Assets

By subscribing to the TimeManager observer pattern, I was able to change the state of various objects based on the time of day. To accomplish this, 'EnableDisableTimeOfDay.cs' was created. This script allows a developer to assign a time of day to an object, then a list of objects to enable or disable based on the time of day. This can be shown duplicated (Figure 13), with two separate GameObjects listed, to have separate object states for various times of day. This allows for object state change with minimal performance overhead. Additionally, with such a large number of entities being changed, the typical method would be to have separate scenes - One for each time of day. With this scenario, I was able to minimize scene transitions, which typically have a jarring effect on a user as the display would freeze when loading in these assets.



```
Enable Disable Time Of Day (Script)
Script              EnableDisableTimeOfDay
Time Manager        TimeManager (TimeManager)
Active Time         Night
Objects To Check
    Size            2
    Element 0       Gate_Open
    Element 1       Gate_Open
Enable Disable Time Of Day (Script)
Script              EnableDisableTimeOfDay
Time Manager        TimeManager (TimeManager)
Active Time         Day
Objects To Check
    Size            2
    Element 0       Gate_Closed
    Element 1       Gate_Closed
```

*Figure 13. The EnableDisableTimeOfDay script being used to change the state of a gate.*

#### 6.4.1.2. UI Elements

As UI elements, such as unit health, are shown in world space due to limitations of VR, this resulted in elements not being displayed correctly if the player was not facing the unit. To account for this, ‘FaceCamera’ was created, which makes the attached GameObject always face the main camera, a part of the Player prefab.

#### 6.4.1.3. Win/Lose condition

Rather than have the player reload the scene or load the menu scene whenever a unit is destroyed, the units remaining are checked.  Depending on which factions units are remaining at the end of the battle, a win or lose condition is activated. This gives them a different end game message before they return to the main menu.

#### 6.4.1.4. Teleportation

To move between areas, TeleportToArea was created. This script could be used in two ways, using method overloading. Firstly, doors within the scene had a box collider attached to them, set to be used as a trigger. This script could then be called using the OnTriggerEnter method, which used the collider given to the trigger as a parameter to EnterRegion, and used as the target of this script. Conversely, if the method was given no parameters, it is assumed that the target is the player singleton, and this is used as the target GameObject. This allows both a direct teleport, through coming into contact with doors, and teleportation through script, which is used within both the graveyard guard and game mediation scripts.

### 6.4.2. Audio

In virtual reality environments, the emphasis is on creating an immersive experience for the user, of which in media, audio typically plays a large role in video games (Sander, 2010). To account for this, 3D audio was used throughout the game world to create binaural audio, with the intent to replicate the way audio is perceived by humans. Ambient sound was utilized, rather than a traditional soundtrack, due to how sound is shown in real life situations. Ordinarily, people do not have a soundtrack, but experience ambient sounds, such as birds, cars and human interaction. Additionally, when considering a virtual reality environment, a soundtrack could overwhelm the user, resulting in diverting the user's attention from gameplay mechanics.

# 7. End-project report

## 7.1. M.V.P and M.A.P Evaluation

When reconsidering the original M.V.P. of the problem:

- The project should have a full game loop where the player can:
    - Dig at a grave to unearth a body or body part.
    - Create a minion using these body parts.
    - Use these minions to take part in a tower-defence style minigame.
- The player should encounter and be able to create several types of units, including a basic combat triangle.
- The player should be able to interact with the villagers outside of the tower-defence minigame for hints and dialogue options.
- This should all occur within virtual reality, with support for the Oculus Rift S.

All of these M.V.P. requirements have been met, albeit with the combat triangle being functionally of very limited use within gameplay.

Furthermore, when considering the M.A.P. requirements, there was partial completion. The graveyard guard uses sight to search for the player, and each limb is counted separately towards calculating the units class.

## 7.2. Project Goal - Learning in a new area

The main reason for the project was to develop the author's experience of working with a new technology stack while employing previously learned techniques. Through this project, the author was able to explore both SteamVR and the Oculus SDK, industry-standard solutions to virtual reality within the Unity Engine, and analyse which solution would be more applicable. For this, several comparisons were made before choosing a technology, which are visualised here.

| Platform | SteamVR | Oculus SDK | OpenVR |
|---|---|---|---|
| Supported HMD's/Target audience | Not vendor specific | Oculus Rift S, Oculus Quest, Oculus Go | Not vendor specific |
| Documentation | Derived from OpenVR and has API level documentation | Minimal documentation | API level documentation |
| Future compatibility | Only requires updating controls to allow for new HMD's | Currently requires a complete rewrite of code | Only requires updating controls to allow for new HMD's |
| Dependencies | Steam client for development, not for running application | Oculus Home | Steam client |
| Store limitations | Not allowed on Oculus Store | No limitations | Not allowed on Oculus Store |

*Table 1. Comparison of various VR SDK's.*

## 7.3. Criticisms

Storing data in plaintext text format, such as the XML type that was used within this project is typically frowned upon within most security contexts. Additionally, as there is no fallback method for generating text data, if a user edited the data and did not conform to the XML structure, it could cause the application to incorrectly assign data or use the placeholder text. However, in this situation, the only data stored within the file are strings for names and hints. With this being in plaintext, it allows a player to modify the content for various reasons, e.g localisation.

SteamVR was relied upon heavily throughout this project for virtual reality interactions, due to their already heavily integrated VR controller. Ideally, using OpenVR and creating my own interaction system would have been ideal when considering the goal of this project.

Third-party visual assets were heavily used during this project, which led to some problems implementing shaders and emissive lighting. In hindsight, keeping the available area smaller and utilizing space more effectively, would have reduced the amount of assets required to be created, making it more plausible for me to create during the three month time period, while also positively impacting frame rate during gameplay due to the lesser number of vertices within the scene. Additionally, this would also have allowed greater control of visual design goals within the project.

# 8. Project post-mortem

## 8.1. Project Objectives

When originally designing the project objectives, they should have been developed using S.M.A.R.T. objectives. As a result of this, they would have become more clear and easily distinguishable when finished and could have driven the project backlog in terms of tasks that require completion.
In regards to the objectives themselves, it would have been preferable for them to be split into further tasks to drive the backlog further. When originally designing the M.A.P. I believed that adding sight or sound to the graveyard guard would not have been as complex as it was, and defining these points further would have led to a more realistic set of project objectives.

## 8.2. Product Specification

The project started out being defined as an anti-tower defence style game, due to my own naivety surrounding this game genre. Upon further research, it was found that the style of gameplay, while it took elements and tropes from both role playing games and tower defence style games, was more similar to typical RPG games.

## 8.3. Development Process

The navigation system was simple, and the way developers have to interact with this system is somewhat convoluted, as you have to create and place the nav point before assigning it within the Navigation Manager script. Ideally, this could have been moved into a custom property drawer and made easier for further development to occur. However, as the focus for this project was not on navigation, and typically in tower defence games, navigation is simplistic too, this would be less important for future development.

Throughout development, editor tools and scripts were created to either aid development, or test various functions.
Due to using the demo scene from Synty's Knights pack as a starting point for my scene, the scene was a mess of GameObjects. Due to this, 'Reparenter' was created. This tool takes the currently selected GameObjects and can either remove their parent GameObject, or add them to a new GameObject. This was used to group GameObjects based on functions - I.E, objects which formed part of a house, were moved into an empty GameObject called a house.
DayNightCycler was a simple script that switched the skybox from day to night and activated or deactivated global lighting based on which was last active. Through this, TimeChanger was created, which let me manually change the time of day within the game to test time specific functions.
ShowContentsOfArrayAsGizmo is a simple script that lets you assign an array, which is then shown as a gizmo within the Unity Editor. This was used to show navigation paths.

## 8.4. Project Management Approach

The original approach to the project took place as weekly sprints using Scrum, though later into the project as features became more complex, sprints became fortnightly. One of the main benefits to using an agile methodology such as Scrum, is being able to look at your process at the end of each sprint and optimize your approach to managing the project. Due to only having a single developer on the team, the three main points of agile artifacts - The review, retrospective and planning meeting, were unable to take place and therefore the continued optimization of the process was minimal at best.

## 8.5. Technologies

Unity does not naturally support serialization of List<T> and due to this, visualization of lists within the editor was limited. To circumvent this, lists were kept private but typically corresponded to an array that could be shown in the editor. However, it is possible to expand Unity's serialization, or create a custom inspector script which will show it. These options would have been better, though due to the M.V.P only requiring a single map, these were left as part of the M.A.P.

Additionally, the Unity editor requires all scripts to be reloaded whenever a script is changed. In smaller projects, this is not an issue, however with the large number of scripts present due to both the Oculus SDK and the SteamVR implementations, this has resulted in compile times being upwards of sixty seconds. This is primarily due to Unity using only a single thread when reloading scripts, and coincidentally, entering play mode suffers the same issues, though apparently this has been fixed in the beta branch of Unity version 2020.

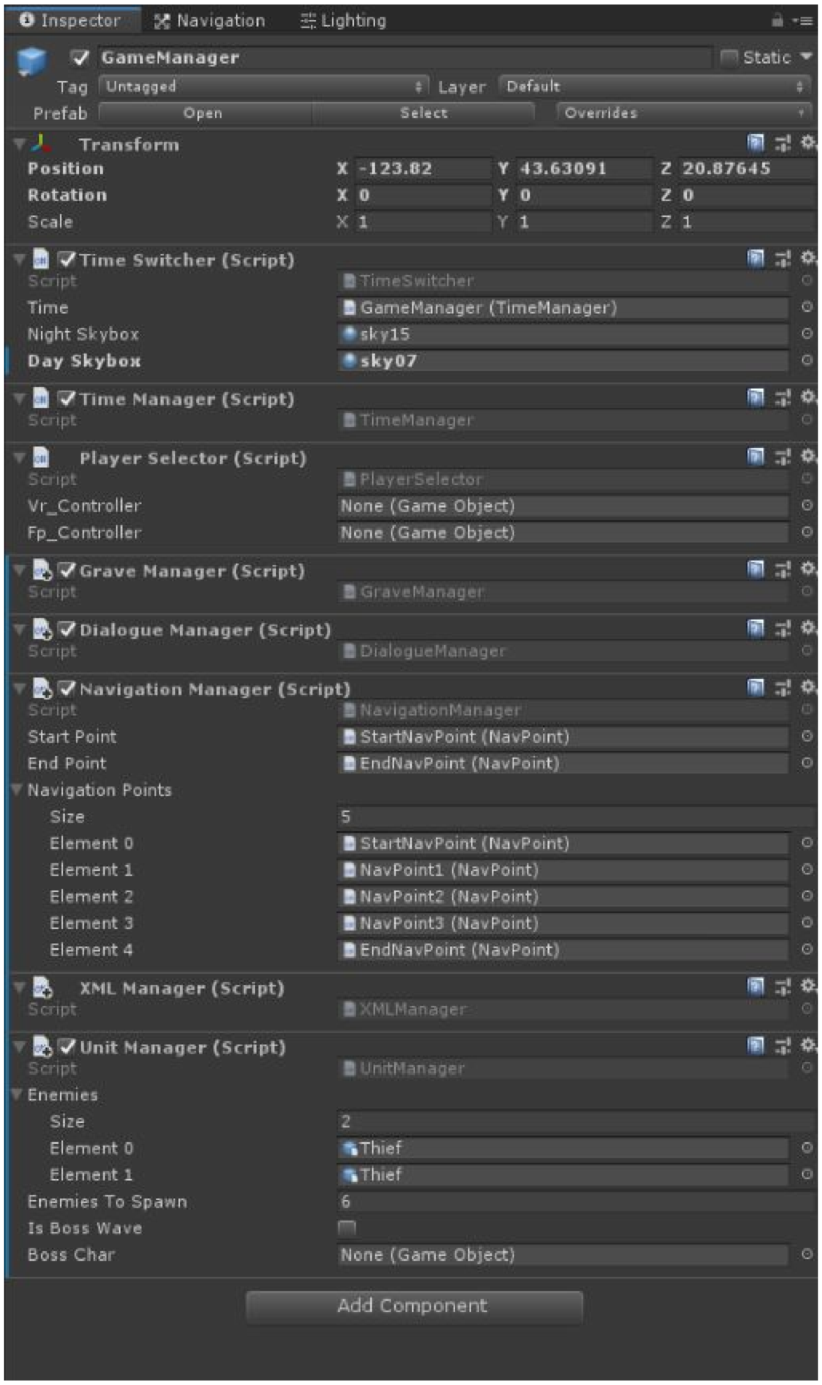
Due to starting the project on the standard render pipeline and the large amount of imported assets, it was infeasible to create shaders using Unity's visual scripting platform, Shader Graph. Additionally, due to the authors lack of experience with creating shaders normally, the overall visual aesthetic of the project was consistently flat, due to the low poly art style.

While SteamVR did allow for support for various devices, they needed to create a level of abstraction to account for this that makes their codebase relatively difficult to understand. Additionally, the complexity of their codebase resulted in not finding features, such as the utility function, SteamVR_Fade, until I had developed my own variation of a screen fade.

## 8.6. Own Performance

For a time during the beginning of the project, I was stuck on completing a single task - Digging at a mesh. I found several tutorials that had similar effects, however, I could not get this to work. This used up a lot of time that could have been used for further development, that was only a small part of the game. In hindsight, I should have moved on from there far quicker.
Additionally, I have relied heavily on singletons within this project. A GameManager, (Figure 14), has several manager scripts with many of them using singletons. This violates the single responsibility principle of SOLID and is widely considered an anti-pattern.
Additionally, the implementation of pathfinding within the project is incredibly naive, and has some bugs with it. If this project was to become commercially viable and not be for a university project, it is likely that Unity's navmesh would have been implemented, or a traditional pathfinding algorithm been implemented manually.
Regrettably, I was unable to conform to my set commit guidelines for GitHub and my usage of Git was suboptimal. I typically prefer to commit for every method creation or significant change. However, this was not kept within this project as is evident from the low number of commits. This would be disastrous if data had been lost or the project had needed to be reverted back to an earlier commit.

While the project was intended to have multiple levels as part of its M.A.P., upon further reflection, not enough thought went into making it easier for development of further levels. While much of the work that has been done would be transferable, it would be difficult to make it significantly different enough without just changing the variety of units that can spawn or the overall aesthetic of the play area.


Inspector Navigation Lighting

GameManager Static

| | |
|---|---|
| Tag | Untagged |
| Layer | Default |
| Prefab | Open / Select / Overrides |

**Transform**

| | | | |
|---|---|---|---|
| Position | X -123.82 | Y 43.63091 | Z 20.87645 |
| Rotation | X 0 | Y 0 | Z 0 |
| Scale | X 1 | Y 1 | Z 1 |

**Time Switcher (Script)**

| | |
|---|---|
| Script | TimeSwitcher |
| Time | GameManager (TimeManager) |
| Night Skybox | sky15 |
| Day Skybox | sky07 |

**Time Manager (Script)**

| | |
|---|---|
| Script | TimeManager |

**Player Selector (Script)**

| | |
|---|---|
| Script | PlayerSelector |
| Vr_Controller | None (Game Object) |
| Fp_Controller | None (Game Object) |

**Grave Manager (Script)**

| | |
|---|---|
| Script | GraveManager |

**Dialogue Manager (Script)**

| | |
|---|---|
| Script | DialogueManager |

**Navigation Manager (Script)**

| | |
|---|---|
| Script | NavigationManager |
| Start Point | StartNavPoint (NavPoint) |
| End Point | EndNavPoint (NavPoint) |
| Navigation Points | |
| Size | 5 |
| Element 0 | StartNavPoint (NavPoint) |
| Element 1 | NavPoint1 (NavPoint) |
| Element 2 | NavPoint2 (NavPoint) |
| Element 3 | NavPoint3 (NavPoint) |
| Element 4 | EndNavPoint (NavPoint) |

**XML Manager (Script)**

| | |
|---|---|
| Script | XMLManager |

**Unit Manager (Script)**

| | |
|---|---|
| Script | UnitManager |
| Enemies | |
| Size | 2 |
| Element 0 | Thief |
| Element 1 | Thief |
| Enemies To Spawn | 6 |
| Is Boss Wave | ☐ |
| Boss Char | None (Game Object) |

Add Component

*Figure 14. The components listed within the GameManager GameObject.*

# 9. Conclusions

The project was able to complete all goals set out at the beginning of the project as its M.V.P. requirements, and introduced a few elements of the M.A.P. In this regard, the project has been successful.
Additionally, the overall goal of this project was to develop the authors ability to adapt to a new technology and critically evaluate various solutions to this technology. To this end, the project can be considered a success.

On the other hand, various elements of the project such as the pathfinding and AI elements have naive implementations and with further developments, would be the first changes I would make. Additionally, more research into what makes this game genre fun would have been helpful, as during the project, I was focused on delivering features and making sure their implementations were correct more than I considered how fun the gameplay was. Typically this would have been dealt with through user testing, however due to COVID-19 and the relatively low number of people with VR headsets, specifically the Oculus Rift S, this was unable to take place.
I would also like to add various types of locomotion to the project, as this would incur relatively little development time per method, and allow a greater number of users to interact with the project successfully.

The development of this project has led to greater understanding of development practices within VR and optimization techniques within this. Additionally, while conducting research for this project, it became clear how viable VR is within software development, from having roots in military training simulations, to more innocuous prospects such as educating surgeons without live cadavers.
In conclusion, the project can be considered a success in multiple areas and allowed the author to expand his knowledge in an emergent technology.

# 10. Reference list

11 bit studios. (2010). *Anomaly Warzone Earth - gameplay video - part 1*. [Online]. [Accessed 24 May 2020]. Available from: https://www.youtube.com/watch?v=j_d4d58u-Lw
Adobe. (2008). *“Mixamo Animations”. Available: https://www.mixamo.com/*

Asbjørn Thirslund. (2017). *How to create a Dialogue System in Unity*. [Online]. [Accessed 11 April 2020]. Available from: https://www.youtube.com/watch?v=_nRzoTzeyxU&t=833s
Atlassian (2011) Trello [Product]. Available at https://trello.com/ (Downloaded: 24 May 2020).

Bethesda Softworks (2017) *Fallout 4 VR* [Computer game]. Available at https://store.steampowered.com/app/611660/Fallout_4_VR/ (Downloaded: 24 May 2020).

Bethesda Softworks (2017) *The Elder Scrolls V: Skyrim VR* [Computer game]. Available at https://store.steampowered.com/app/611670/The_Elder_Scrolls_V_Skyrim_VR/ (Downloaded: 24 May 2020).

Chekhov, Anton as quoted in “Sergei Shchukin (1911) *The memoirs of Shchukin,* unpaginated

Christou, C. G., & Aristidou, P. (2017). *Steering Versus Teleport Locomotion for Head Mounted Displays. Augmented Reality, Virtual Reality, and Computer Graphics* Available at: https://link.springer.com/chapter/10.1007%2F978-3-319-60928-7_37 (Downloaded 24 May 2020)

Code First. (2019). *C# Observer Pattern Example.* [Online]. [Accessed 11 April 2020]. Available from: https://www.youtube.com/watch?v=oVAFvyICmbw
Corey Ladas, 2008, *Scrum-ban*, Viewed 21 April 2020,

Crute, A., 2019. *Recording And Mixing Audio For Virtual Reality | Musictech*. [online] MusicTech. Available at: https://www.musictech.net/features/trends/vr-recording-and-mixing [Accessed 15 May 2020].
Cyreal. (2016). “*Adamina*”. Available: https://fonts.google.com/specimen/Adamina
Devolver Digital (2017) *GORN* [Computer game]. Available at https://store.steampowered.com/app/578620/GORN/ (Downloaded: 24 May 2020).

ElectricNightOwl (2017) *Arm Swinger* [Product]. Available at https://github.com/ElectricNightOwl/ArmSwinger (Downloaded: 24 May 2020).

Epic Games (2017) *Robo Recall* [Computer game]. Available at https://www.epicgames.com/roborecall/en-US/home (Downloaded: 21 April 2020).

Huiberts, S., 2010. *Captivating Sound The Role Of Audio For Immersion In Computer Games*. University of Portsmouth. Available at: https://download.captivatingsound.com/Sander_Huiberts_CaptivatingSound.pdf [Accessed 24 May 2020].

Jeff Rosen, John Graham (2010) *Humble Bundle* [Product]. Available at https://www.humblebundle.com/ (Accessed: 24 May 2020).

Johnson, David M. (2005) *Introduction to and Review of Simulator Sickness Research* , Publisher: U.S Army Research Institute, Arlington, VA, USA Available at: https://pdfs.semanticscholar.org/9ed8/8e0e36ac99ce49fa3386d632acb27ca7a8dd.pdf(Downloaded 24 May 2020)

Ken Schwaber & Jeff Sutherland,  1995, *SCRUM Development Process* , viewed 21 April 2020, http://agilix.nl/resources/scrum_OOPSLA_95.pdf

Kohsuke Kawaguchi (2011) *Jenkins*[Product]. Available at https://www.jenkins.io/ (Downloaded: 24 May 2020).
Linus Torvalds (2005) *Git* [Product]. Available at https://git-scm.com/ (Downloaded: 24 May 2020).

McCullough, M, Xu, H, Michaelson, J, Jackoski, M, Pease, W, Cobb, W, Kalescky, W, Ladd, J, Williams, B. 2015. *Myo arm: swinging to explore a VE* [online] MusicTech. Available at: https://dl.acm.org/doi/10.1145/2804408.2804416 [Accessed 15 May 2020].

Mgsvevo. (2015). "*Classic Skybox*". Available: https://assetstore.unity.com/packages/2d/textures-materials/sky/classic-skybox-24923

Ninja Kiwi (2007) *Bloons Tower Defence* [Computer game]. Available at https://ninjakiwi.com/Games/Tower-Defense/Bloons-Tower-Defense.html (Downloaded: 24 May 2020).

Ninja Kiwi (2018) *Bloons TD 6* [Computer game]. Available at https://store.steampowered.com/app/960090/Bloons_TD_6/ (Downloaded: 24 May 2020).

Oculus (2020).*Oculus Guardian System*[online] Oculus. Available at: https://developer.oculus.com/documentation/native/android/mobile-guardian/ [Accessed 24 May 2020].

Rigney, R. 2013. *Even the Best Tower Defence Games Are Just Plain Boring*. [online] wired.com. Available at: https://www.wired.com/2013/11/anomaly-2/   [Accessed 24 May 2020].
Serafin & Serafin (2004). *SOUND DESIGN TO ENHANCE PRESENCE IN PHOTOREALISTIC VIRTUAL REALITY* Available from: https://smartech.gatech.edu/bitstream/handle/1853/50913/SerafinSera (Downloaded 15 May 2020).

Solum Night. (2017). "*Low Poly Pack - Environment Lite*". Available: https://assetstore.unity.com/packages/3d/props/exterior/low-poly-pack-environment-lite-102039

Steam  (2014).*SteamVR Chaperone FAQ* [online] Valve. Available at: https://support.steampowered.com/kb_article.php?ref=6281-TOKV-4722 [Accessed 24 May 2020].

Steam  (2014).*SteamVR* [online] Valve. Available at: https://store.steampowered.com/news/?appids=250820&enddate=1420070400&appgroupname=SteamVR [Accessed 15 May 2020].

Steam  (2020). *Anomaly: Warzone Earth*. [online] Valve. Available at: https://store.steampowered.com/app/91200/Anomaly_Warzone_Earth/  [Accessed 15 May 2020].
Survios (2018) *Creed: Rise to Glory VR*  [Computer game]. Available at https://store.steampowered.com/app/804490/Creed_Rise_to_Glory/ (Downloaded: 24 May 2020).

Synty Studios. (unknown). *"Polygon Adventure". Available: https://syntystore.com/products/polygon-adventure-pack*

Synty Studios. (unknown). *"Polygon Knights". Available:https://syntystore.com/products/polygon-knights-pack*

Synty Studios. (unknown). *"Polygon Vikings". Available: https://syntystore.com/products/polygon-vikings-pack*

Synty Studios. (unknown). *"Simple Dungeons". Available: https://syntystore.com/products/simple-dungeons-cartoon-assets*

Taiichi Ohno (1988). Toyota Production System: Beyond Large-Scale Production

Tom Preston-Werner, Chris Wanstrath, P.J. Hyett, Scott Chacon (2008) GitHub [Product]. Available at https://github.com/ (Downloaded: 24 May 2020).

Tsingos, N., 2016. *Audio Challenges In Virtual And Augmented Reality*. Available at: https://www.youtube.com/watch?v=z9P4JbKj7mk

Valve. 2020. *Steam Hardware & Software Survey: April 2020.* [online] Medium. Available at: https://store.steampowered.com/hwsurvey/Steam-Hardware-Software-Survey-Welcome-to-Steam  [Accessed 24 May 2020].

bradweiers (2018), as quoted in Wekthor. *OpenVR vs SteamVR*. [online] Unity Forums. Available at: https://forum.unity.com/threads/openvr-vs-steamvr.516955/ [Accessed 24 May 2020].

https://www.developmentthatpays.com/scrumban

u/DynoBeast (2018). *Why do games not use GORN style movement more?*. [online] Reddit. Available at: https://www.reddit.com/r/Vive/comments/7hjcnl/why_do_games_not_use_gorn_style_movement_more/ [Accessed 15 May 2020].

u/Kamsides (2019), as quoted in u/aysayaa. *Are you okay with having multiple game libraries/launcher? Or are you only using one...*. [online] Reddit. Available at: https://www.reddit.com/r/gaming/comments/a7bnux/question_are_you_okay_with_having_multiple_game/ [Accessed 24 May 2020].

u/Rtza (2018). *Why do games not use GORN style movement more?*. [online] Reddit. Available at: https://www.reddit.com/r/Vive/comments/7hjcnl/why_do_games_not_use_gorn_style_movement_more/ [Accessed 15 May 2020].
u/TopinambourSansSel (2018). *I'm in deep love with Armswinger*. [online] Reddit. Available at: https://www.reddit.com/r/H3VR/comments/7o586k/im_in_deep_love_with_armswinger/ [Accessed 15 May 2020].

u/lemonlemons (2018), as quoted in u/BL24L. *Smooth locomotion vs Teleportation for immersion*. [online] Reddit. Available at: https://www.reddit.com/r/oculus/comments/97zll7/smooth_locomotion_vs_teleportation_for_immersion/ [Accessed 15 May 2020].

u/rust_anton (2018), as quoted in u/TopinambourSansSel. *I'm in deep love with Armswinger*. [online] Reddit. Available at: https://www.reddit.com/r/H3VR/comments/7o586k/im_in_deep_love_with_armswinger/ [Accessed 15 May 2020].

ustwo, g., 2015. *Designing Sound For Virtual Reality*. [online] Medium. Available at: https://medium.com/@ustwogames/designing-sound-for-virtual-reality-a37a40e80463 [Accessed 15 May 2020].

# 11. Bibliography

Beat Games (2019) Beat Saber. Available at
https://store.steampowered.com/app/620980/Beat_Saber/ (Downloaded: 24 May 2020).
Dużmańska, N, Strojny, P, Strojny, A., 2018. *Can Simulator Sickness Be Avoided? A Review on Temporal Aspects of Simulator Sickness* [online] FrontierSin. Available at:
https://www.frontiersin.org/articles/10.3389/fpsyg.2018.02132/full [Accessed 15 May 2020].

Flick. [no date]. *Mesh Deformation Making a Stress Ball*. [Online]. [22 May 2020]. Available from: https://catlikecoding.com/unity/tutorials/mesh-deformation/

HTC. *Setting up a room-scale play area.* [online] vive.com. Available at:
https://www.vive.com/eu/support/vive-pro-hmd/category_howto/setting-up-room-scale-play-area.html [Accessed 24 May 2020].

LetsPlayKongregate. (2011). *Let's Play: Villainous - Part 1 [The Basics]*. [Online]. [Accessed 24 May 2020]. Available from: https://www.youtube.com/watch?v=fTNr-LqZ0w0

Neat Corporation (2018) Budget Cuts. Available at
https://store.steampowered.com/app/400940/Budget_Cuts/ (Downloaded: 24 May 2020).

Ninja Theory (2018) Hellblade: Senua's Sacrifice VR Edition. Available at
https://store.steampowered.com/app/747350/Hellblade_Senuas_Sacrifice_VR_Edition/
(Downloaded: 24 May 2020).
Oculus. *Oculus Guardian System*. [online] developer.oculus.com. Available at:
https://developer.oculus.com/documentation/native/pc/dg-guardian-system/ [Accessed 24 May 2020].

PlayStation. (2018). *Creed: Rise to Glory - PS VR Gameplay | PlayStation Underground*. [Online]. [Accessed 24 May 2020]. Available from:
https://www.youtube.com/watch?v=r6zIAT3H0VQ

Rede (2017) Villainous. Available at https://www.kongregate.com/games/rete/villainous
(Downloaded: 24 May 2020).

SUPERHOT Team (2017) Superhot VR. Available at
https://store.steampowered.com/app/617830/SUPERHOT_VR/ (Downloaded: 24 May 2020).

UploadVR. (2017). *Skyrim VR (Bethesda) PSVR Move Controller Smooth Locomotion Gameplay*. [Online]. [Accessed 24 May 2020]. Available from: https://www.youtube.com/watch?v=PLbykFn4Q5k&list=PLDXBKX3q25UzFoks7FdsRH9q-cKvvrXxj&index=97

White, P. 2018. *Lessons Learned From Five Years of VR Locomotion Experiments*. [online] developer.oculus.com. Available at: https://www.gamasutra.com/blogs/PaulWhite/20180420/316170/Lessons_Learned_From_Five_Years_of_VR_Locomotion_Experiments.php [Accessed 24 May 2020].

Xinreality. *Locomotion Methods*. [online] xinreality.com. Available at: https://xinreality.com/wiki/Locomotion [Accessed 24 May 2020].

jacksepticeye (2020) *Fighting Zombies in VR is TERRIFYING* | *The Walking Dead Saints and Sinners VR #1* . Available at: https://www.youtube.com/watch?v=Hd6fiBqOi7E (Accessed: 22 May 2020).

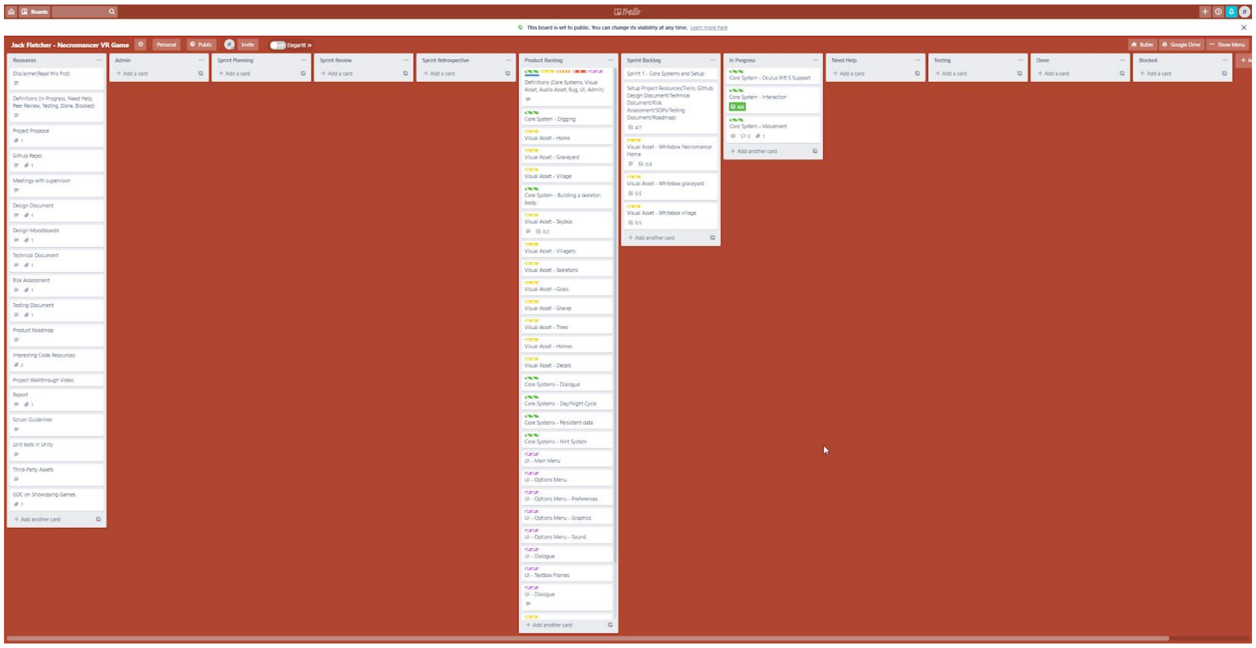
u/KeeganR Ms (2018). *Smooth Locomotion: Does it make you feel ill?*. [online] Reddit. Available at: https://www.reddit.com/r/Vive/comments/7iz1m6/smooth_locomotion_does_it_make_you_feel_ill/ [Accessed 24 May 2020].

# 12. Appendices

## 12.1. Section A - Project Management

NB I'm aware that due to the scale of these images, they may not be legible. For this reason, their files are shared at:

https://drive.google.com/open?id=1XPcbIXJUTsqJbEkx4SmYARFsZhfCTDR0



This board is set to public. You can change its visibility at any time. Learn more here

Jack Fletcher - Necromancer VR Game

Resources
- Disclaimer(Read this first)
- Definitions (In Progress, Need Help, Peer Review, Testing, Done, Blocked)
- Project Proposal 1
- Github Repo: 1
- Meetings with supervisor:
- Design Document 1
- Design Moodboards 1
- Technical Document 1
- Risk Assessment 1
- Testing Document 1
- Product Roadmap
- Interesting Code Resources: 2
- Project Walkthrough Video
- Report 1
- Scrum Guidelines
- Unit tests in Unity
- Third-Party Assets
- GDC on Showcasing Games 1

Admin

Sprint Planning

Sprint Review

Sprint Retrospective

Product Backlog
- Definitions (Core Systems, Visual Asset, Audio Asset, Bug, UI, Admin)
- Core System - Digging
- Visual Asset - Home
- Visual Asset - Graveyard
- Visual Asset - Village
- Core System - Building a skeleton body.
- Visual Asset - Skybox 0/2
- Visual Asset - Villagers
- Visual Asset - Skeletons
- Visual Asset - Grass
- Visual Asset - Graves
- Visual Asset - Trees
- Visual Asset - Homes
- Visual Asset - Decals
- Core Systems - Dialogue
- Core Systems - Day/Night Cycle
- Core Systems - Persistent data
- Core Systems - Hint System
- UI - Main Menu
- UI - Options Menu
- UI - Options Menu - Preferences
- UI - Options Menu - Graphics
- UI - Options Menu - Sound
- UI - Dialogue
- UI - Textbox Frames
- UI - Dialogue

Sprint Backlog
- Sprint 1 - Core Systems and Setup
- Setup Project Resources(Trello, Github, Design Document/Technical Document/Risk Assessment/SOPs/Testing Document/Roadmap) 4/7
- Visual Asset - Whitebox Necromancer Home 0/4
- Visual Asset - Whitebox graveyard 0/5
- Visual Asset - Whitebox village 0/5

In Progress
- Core System - Oculus Rift S Support
- Core System - Interaction 4/4
- Core System - Movement 3 1

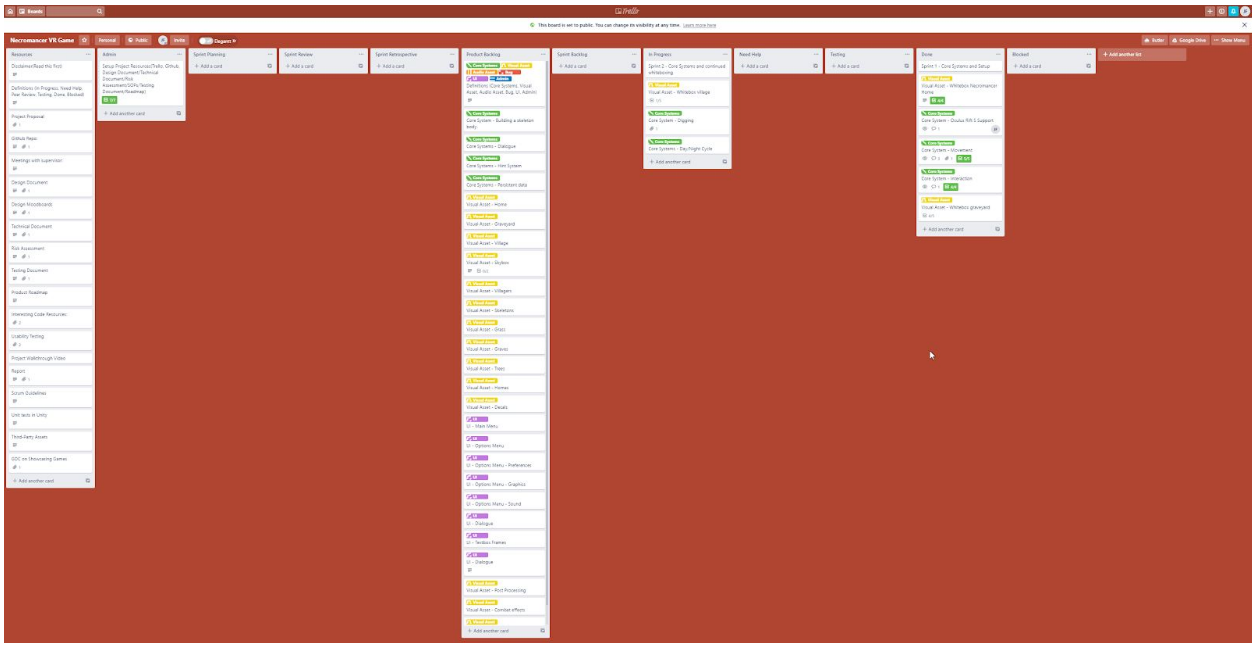
Need Help

Testing

Done

Blocked

Figure 15. Week 1 Trello submission.



This board is set to public. You can change its visibility at any time. Learn more here

Necromancer VR Game

Resources
- Disclaimer(Read this first)
- Definitions (In Progress, Need Help, Peer Review, Testing, Done, Blocked)
- Project Proposal 1
- Github Repo: 1
- Meetings with supervisor:
- Design Document 1
- Design Moodboards 1
- Technical Document 1
- Risk Assessment 1
- Testing Document 1
- Product Roadmap
- Interesting Code Resources: 2
- Usability Testing 2
- Project Walkthrough Video
- Report 1
- Scrum Guidelines
- Unit tests in Unity
- Third-Party Assets
- GDC on Showcasing Games 1

Admin
- Setup Project Resources(Trello, Github, Design Document/Technical Document/Risk Assessment/SOPs/Testing Document/Roadmap) 7/7

Sprint Planning

Sprint Review

Sprint Retrospective

Product Backlog
- Core Systems · Visual Asset · Audio Asset · Bug · UI · Admin — Definitions (Core Systems, Visual Asset, Audio Asset, Bug, UI, Admin)
- Core Systems — Core System - Building a skeleton body.
- Core Systems — Core Systems - Dialogue
- Core Systems — Core Systems - Hint System
- Core Systems — Core Systems - Persistent data
- Visual Asset — Visual Asset - Home
- Visual Asset — Visual Asset - Graveyard
- Visual Asset — Visual Asset - Village
- Visual Asset — Visual Asset - Skybox 0/2
- Visual Asset — Visual Asset - Villagers
- Visual Asset — Visual Asset - Skeletons
- Visual Asset — Visual Asset - Grass
- Visual Asset — Visual Asset - Graves
- Visual Asset — Visual Asset - Trees
- Visual Asset — Visual Asset - Homes
- Visual Asset — Visual Asset - Decals
- UI — UI - Main Menu
- UI — UI - Options Menu
- UI — UI - Options Menu - Preferences
- UI — UI - Options Menu - Graphics
- UI — UI - Options Menu - Sound
- UI — UI - Dialogue
- UI — UI - Textbox Frames
- UI — UI - Dialogue
- Visual Asset — Visual Asset - Post Processing
- Visual Asset — Visual Asset - Combat effects

Sprint Backlog

In Progress
- Sprint 2 - Core Systems and continued whiteboxing
- Visual Asset — Visual Asset - Whitebox village 1/5
- Core Systems — Core System - Digging 1
- Core Systems — Core Systems - Day/Night Cycle

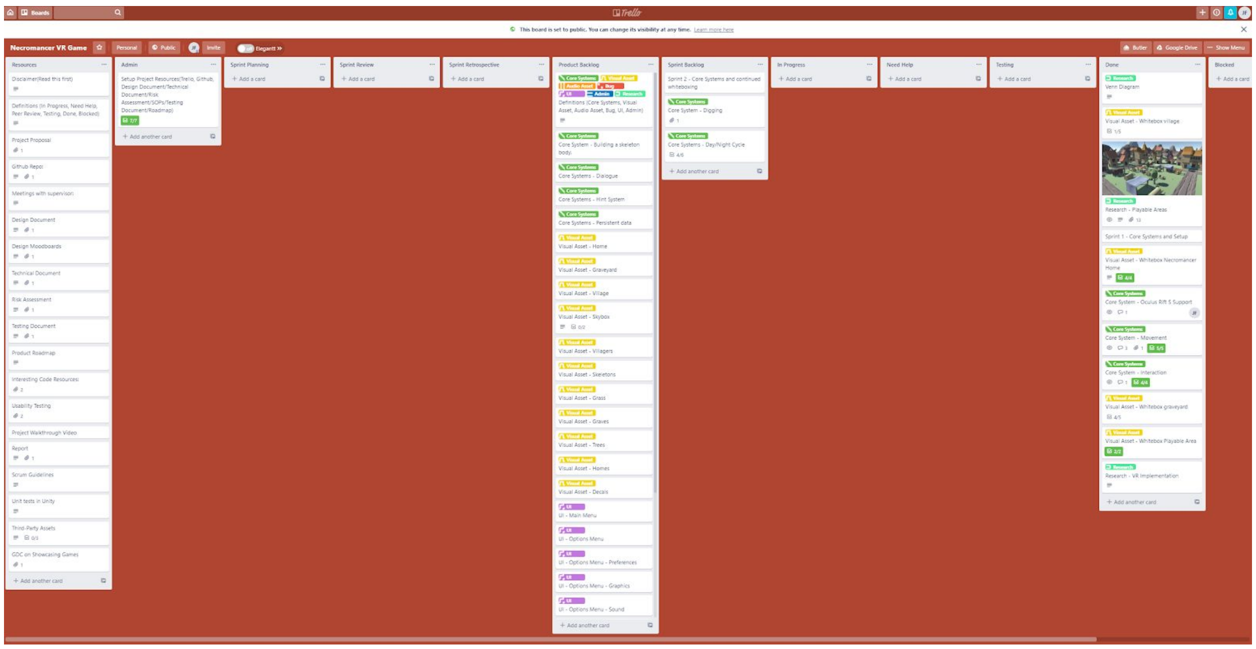
Need Help

Testing

Done
- Sprint 1 - Core Systems and Setup
- Visual Asset — Visual Asset - Whitebox Necromancer Home 4/4
- Core Systems — Core System - Oculus Rift S Support 1
- Core Systems — Core System - Movement 3 1 5/5
- Core Systems — Core System - Interaction 1 4/4
- Visual Asset — Visual Asset - Whitebox graveyard 4/5

Blocked

Figure 16. Week 2 Trello submission.



Necromancer VR Game

This board is set to public. You can change its visibility at any time. Learn more here

**Resources**
- Disclaimer(Read this first)
- Definitions (In Progress, Need Help, Peer Review, Testing, Done, Blocked)
- Project Proposal
- Github Repo:
- Meetings with supervisor:
- Design Document
- Design Moodboards
- Technical Document
- Risk Assessment
- Testing Document
- Product Roadmap
- Interesting Code Resources:
- Usability Testing
- Project Walkthrough Video
- Report
- Scrum Guidelines
- Unit tests in Unity
- Third-Party Assets 0/3
- GDC on Showcasing Games

**Admin**
- Setup Project Resources(Trello, Github, Design Document/Technical Document/Risk Assessment/SOPs/Testing Document/Roadmap) 7/7

**Sprint Planning**

**Sprint Review**

**Sprint Retrospective**

**Product Backlog**
- Core Systems, Visual Asset, Audio Asset, Bug, UI, Admin, Research — Definitions (Core Systems, Visual Asset, Audio Asset, Bug, UI, Admin)
- Core Systems — Core System - Building a skeleton body.
- Core Systems — Core Systems - Dialogue
- Core Systems — Core Systems - Hint System
- Core Systems — Core Systems - Persistent data
- Visual Asset — Visual Asset - Home
- Visual Asset — Visual Asset - Graveyard
- Visual Asset — Visual Asset - Village
- Visual Asset — Visual Asset - Skybox 0/2
- Visual Asset — Visual Asset - Villagers
- Visual Asset — Visual Asset - Skeletons
- Visual Asset — Visual Asset - Grass
- Visual Asset — Visual Asset - Graves
- Visual Asset — Visual Asset - Trees
- Visual Asset — Visual Asset - Homes
- Visual Asset — Visual Asset - Decals
- UI — UI - Main Menu
- UI — UI - Options Menu
- UI — UI - Options Menu - Preferences
- UI — UI - Options Menu - Graphics
- UI — UI - Options Menu - Sound

**Sprint Backlog**
- Sprint 2 - Core Systems and continued whiteboxing
- Core Systems — Core System - Digging
- Core Systems — Core Systems - Day/Night Cycle 4/6

**In Progress**

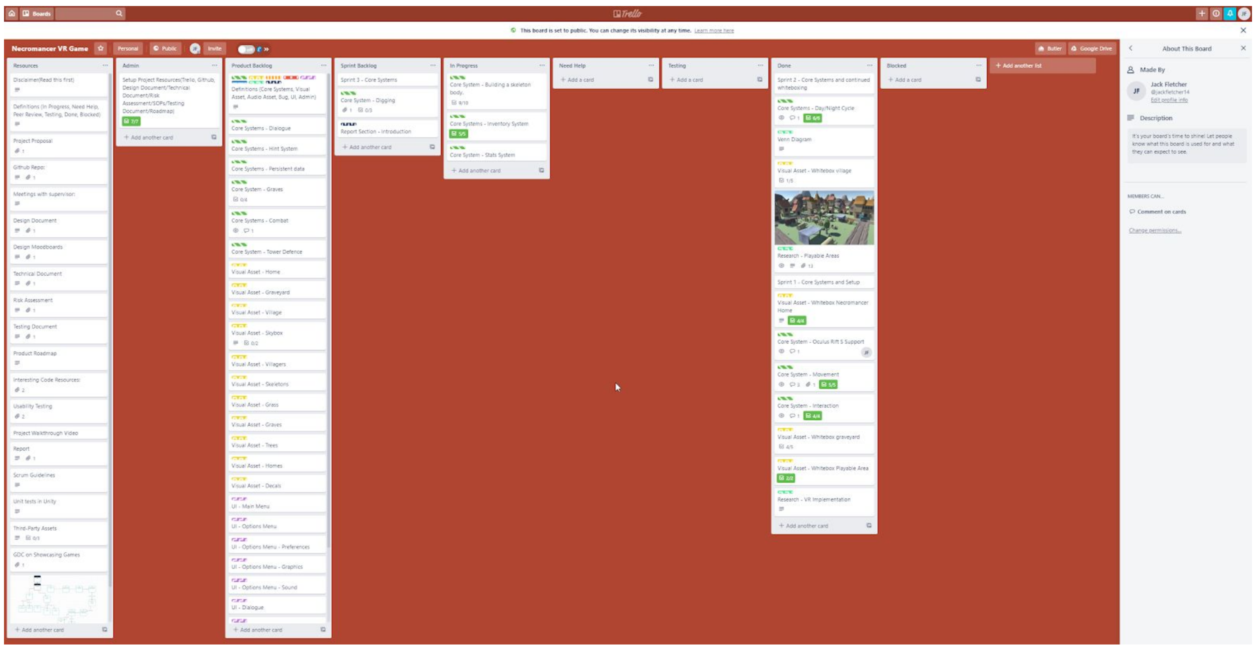
**Need Help**

**Testing**

**Done**
- Research — Venn Diagram
- Visual Asset — Visual Asset - Whitebox village 1/5
- Research — Research - Playable Areas
- Sprint 1 - Core Systems and Setup
- Visual Asset — Visual Asset - Whitebox Necromancer Home 4/4
- Core Systems — Core System - Oculus Rift S Support
- Core Systems — Core System - Movement 5/5
- Core Systems — Core System - Interaction 4/4
- Visual Asset — Visual Asset - Whitebox graveyard 4/5
- Visual Asset — Visual Asset - Whitebox Playable Area 2/2
- Research — Research - VR Implementation

**Blocked**

Figure 17. Week 3 Trello submission



Necromancer VR Game

This board is set to public. You can change its visibility at any time. Learn more here

**Resources**
- Disclaimer(Read this first)
- Definitions (In Progress, Need Help, Peer Review, Testing, Done, Blocked)
- Project Proposal
- Github Repo:
- Meetings with supervisor:
- Design Document
- Design Moodboards
- Technical Document
- Risk Assessment
- Testing Document
- Product Roadmap
- Interesting Code Resources:
- Usability Testing
- Project Walkthrough Video
- Report
- Scrum Guidelines
- Unit tests in Unity
- Third-Party Assets 0/3
- GDC on Showcasing Games

**Admin**
- Setup Project Resources(Trello, Github, Design Document/Technical Document/Risk Assessment/SOPs/Testing Document/Roadmap) 7/7

**Product Backlog**
- Definitions (Core Systems, Visual Asset, Audio Asset, Bug, UI, Admin)
- Core Systems - Dialogue
- Core Systems - Hint System
- Core Systems - Persistent data
- Core System - Graves 0/4
- Core Systems - Combat
- Core System - Tower Defence
- Visual Asset - Home
- Visual Asset - Graveyard
- Visual Asset - Village
- Visual Asset - Skybox 0/2
- Visual Asset - Villagers
- Visual Asset - Skeletons
- Visual Asset - Grass
- Visual Asset - Graves
- Visual Asset - Trees
- Visual Asset - Homes
- Visual Asset - Decals
- UI - Main Menu
- UI - Options Menu
- UI - Options Menu - Preferences
- UI - Options Menu - Graphics
- UI - Options Menu - Sound
- UI - Dialogue

**Sprint Backlog**
- Sprint 3 - Core Systems
- Core System - Digging 0/3
- Report Section - Introduction

**In Progress**
- Core System - Building a skeleton body. 9/10
- Core Systems - Inventory System 5/5
- Core System - Stats System

**Need Help**

**Testing**

**Done**
- Sprint 2 - Core Systems and continued whiteboxing
- Core Systems - Day/Night Cycle 6/6
- Venn Diagram
- Visual Asset - Whitebox village 1/5
- Research - Playable Areas
- Sprint 1 - Core Systems and Setup
- Visual Asset - Whitebox Necromancer Home 4/4
- Core System - Oculus Rift S Support
- Core System - Movement 5/5
- Core System - Interaction 4/4
- Visual Asset - Whitebox graveyard 4/5
- Visual Asset - Whitebox Playable Area 2/2
- Research - VR Implementation

**Blocked**

About This Board

Made By

Jack Fletcher
@jackfletcher14
Edit profile info

Description

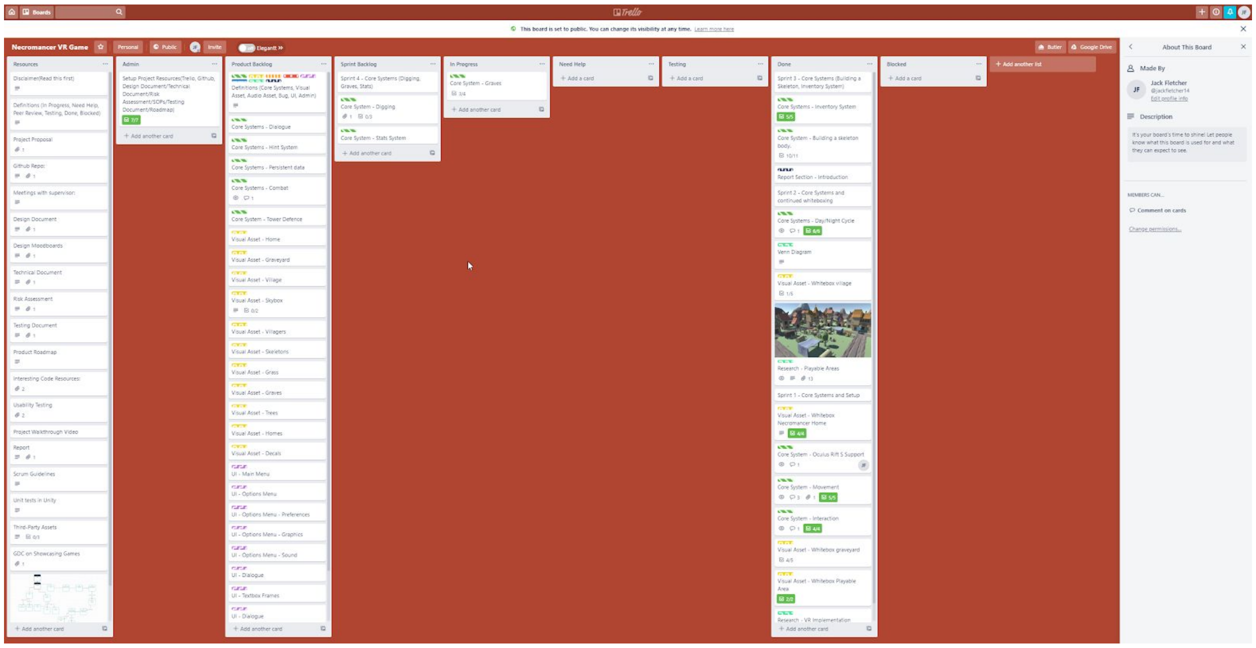
It's your board's time to shine! Let people know what this board is used for and what they can expect to see.

MEMBERS CAN...

Comment on cards

Change permissions...

Figure 18. Week 4 Trello submission.


Boards

Trello

This board is set to public. You can change its visibility at any time. Learn more here

Necromancer VR Game · Personal · Public · Invite · Elegantt · Butler · Google Drive

**Resources**
- Disclaimer(Read this first)
- Definitions (In Progress, Need Help, Peer Review, Testing, Done, Blocked)
- Project Proposal — 1
- Github Repo: — 1
- Meetings with supervisor:
- Design Document — 1
- Design Moodboards — 1
- Technical Document — 1
- Risk Assessment — 1
- Testing Document — 1
- Product Roadmap
- Interesting Code Resources: — 2
- Usability Testing — 2
- Project Walkthrough Video
- Report — 1
- Scrum Guidelines
- Unit tests in Unity
- Third-Party Assets — 0/1
- GDC on Showcasing Games — 1
- + Add another card

**Admin**
- Setup Project Resources(Trello, Github, Design Document/Technical Document/Risk Assessment/SOPs/Testing Document/Roadmap) — 7/7
- + Add another card

**Product Backlog**
- Definitions (Core Systems, Visual Asset, Audio Asset, Bug, UI, Admin)
- Core Systems - Dialogue
- Core Systems - Hint System
- Core Systems - Persistent data
- Core Systems - Combat — 1
- Core System - Tower Defence
- Visual Asset - Home
- Visual Asset - Graveyard
- Visual Asset - Village
- Visual Asset - Skybox — 0/2
- Visual Asset - Villagers
- Visual Asset - Skeletons
- Visual Asset - Grass
- Visual Asset - Graves
- Visual Asset - Trees
- Visual Asset - Homes
- Visual Asset - Decals
- UI - Main Menu
- UI - Options Menu
- UI - Options Menu - Preferences
- UI - Options Menu - Graphics
- UI - Options Menu - Sound
- UI - Dialogue
- UI - Textbox Frames
- UI - Dialogue
- + Add another card

**Sprint Backlog**
- Sprint 4 - Core Systems (Digging, Graves, Stats)
- Core System - Digging — 1 — 0/3
- Core System - Stats System
- + Add another card

**In Progress**
- Core System - Graves — 3/4
- + Add another card

**Need Help**
- + Add a card

**Testing**
- + Add a card

**Done**
- Sprint 3 - Core Systems (Building a Skeleton, Inventory System)
- Core Systems - Inventory System — 5/5
- Core System - Building a skeleton body. — 10/11
- Report Section - Introduction
- Sprint 2 - Core Systems and continued whiteboxing
- Core Systems - Day/Night Cycle — 1 — 6/6
- Venn Diagram
- Visual Asset - Whitebox village — 1/5
- Research - Playable Areas — 13
- Sprint 1 - Core Systems and Setup
- Visual Asset - Whitebox Necromancer Home — 4/4
- Core System - Oculus Rift S Support — 1
- Core System - Movement — 3 — 1 — 5/5
- Core System - Interaction — 1 — 4/4
- Visual Asset - Whitebox graveyard — 4/5
- Visual Asset - Whitebox Playable Area — 2/2
- Research - VR Implementation
- + Add another card

**Blocked**
- + Add a card

+ Add another list

About This Board

Made By

Jack Fletcher
@jackfletcher14
Edit profile info

Description

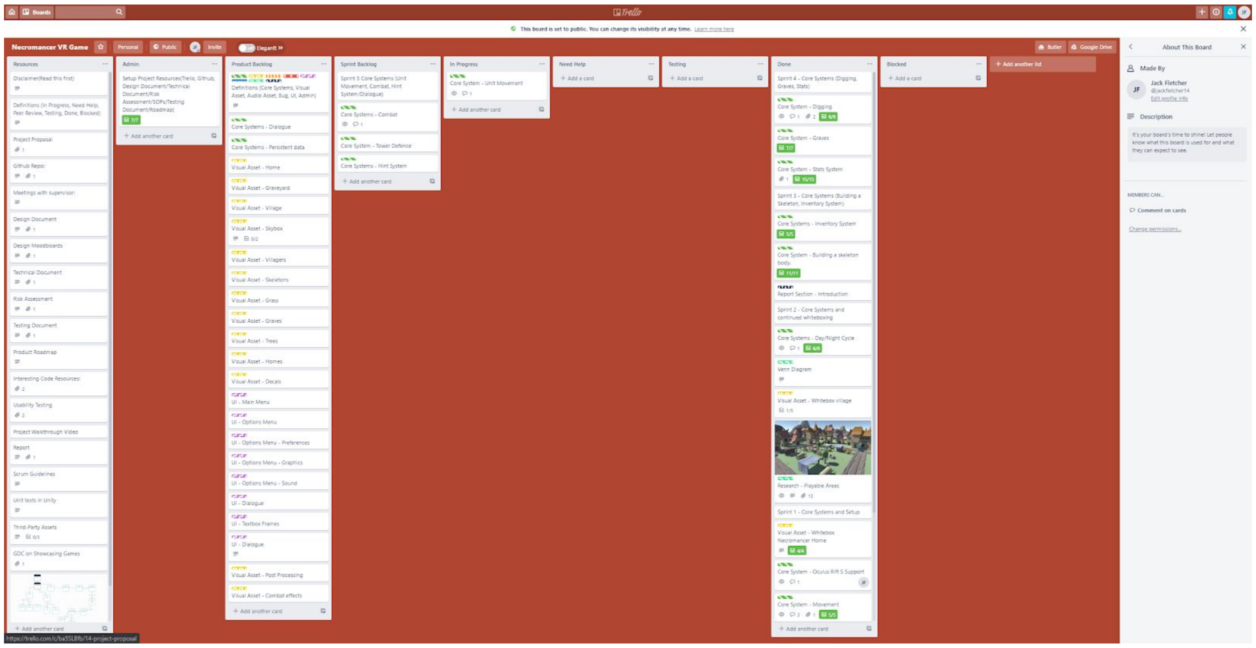
It's your board's time to shine! Let people know what this board is used for and what they can expect to see.

MEMBERS CAN...

Comment on cards

Change permissions...

Figure 19. Week 5 Trello submission.


Boards

Trello

This board is set to public. You can change its visibility at any time. Learn more here

Necromancer VR Game · Personal · Public · Invite · Elegantt · Butler · Google Drive

**Resources**
- Disclaimer(Read this first)
- Definitions (In Progress, Need Help, Peer Review, Testing, Done, Blocked)
- Project Proposal — 1
- Github Repo: — 1
- Meetings with supervisor:
- Design Document — 1
- Design Moodboards — 1
- Technical Document — 1
- Risk Assessment — 1
- Testing Document — 1
- Product Roadmap
- Interesting Code Resources: — 2
- Usability Testing — 2
- Project Walkthrough Video
- Report — 1
- Scrum Guidelines
- Unit tests in Unity
- Third-Party Assets — 0/1
- GDC on Showcasing Games — 1
- + Add another card

https://trello.com/c/ba5SL8fb/14-project-proposal

**Admin**
- Setup Project Resources(Trello, Github, Design Document/Technical Document/Risk Assessment/SOPs/Testing Document/Roadmap) — 7/7
- + Add another card

**Product Backlog**
- Definitions (Core Systems, Visual Asset, Audio Asset, Bug, UI, Admin)
- Core Systems - Dialogue
- Core Systems - Persistent data
- Visual Asset - Home
- Visual Asset - Graveyard
- Visual Asset - Village
- Visual Asset - Skybox — 0/2
- Visual Asset - Villagers
- Visual Asset - Skeletons
- Visual Asset - Grass
- Visual Asset - Graves
- Visual Asset - Trees
- Visual Asset - Homes
- Visual Asset - Decals
- UI - Main Menu
- UI - Options Menu
- UI - Options Menu - Preferences
- UI - Options Menu - Graphics
- UI - Options Menu - Sound
- UI - Dialogue
- UI - Textbox Frames
- UI - Dialogue
- Visual Asset - Post Processing
- Visual Asset - Combat effects
- + Add another card

**Sprint Backlog**
- Sprint 5 Core Systems (Unit Movement, Combat, Hint System/Dialogue)
- Core Systems - Combat — 1
- Core System - Tower Defence
- Core Systems - Hint System
- + Add another card

**In Progress**
- Core System - Unit Movement — 1
- + Add another card

**Need Help**
- + Add a card

**Testing**
- + Add a card

**Done**
- Sprint 4 - Core Systems (Digging, Graves, Stats)
- Core System - Digging — 1 — 2 — 6/6
- Core System - Graves — 7/7
- Core System - Stats System — 1 — 15/15
- Sprint 3 - Core Systems (Building a Skeleton, Inventory System)
- Core Systems - Inventory System — 5/5
- Core System - Building a skeleton body. — 11/11
- Report Section - Introduction
- Sprint 2 - Core Systems and continued whiteboxing
- Core Systems - Day/Night Cycle — 1 — 6/6
- Venn Diagram
- Visual Asset - Whitebox village — 1/5
- Research - Playable Areas — 13
- Sprint 1 - Core Systems and Setup
- Visual Asset - Whitebox Necromancer Home — 4/4
- Core System - Oculus Rift S Support — 1
- Core System - Movement — 3 — 1 — 5/5
- + Add another card

**Blocked**
- + Add a card

+ Add another list

About This Board

Made By

Jack Fletcher
@jackfletcher14
Edit profile info

Description

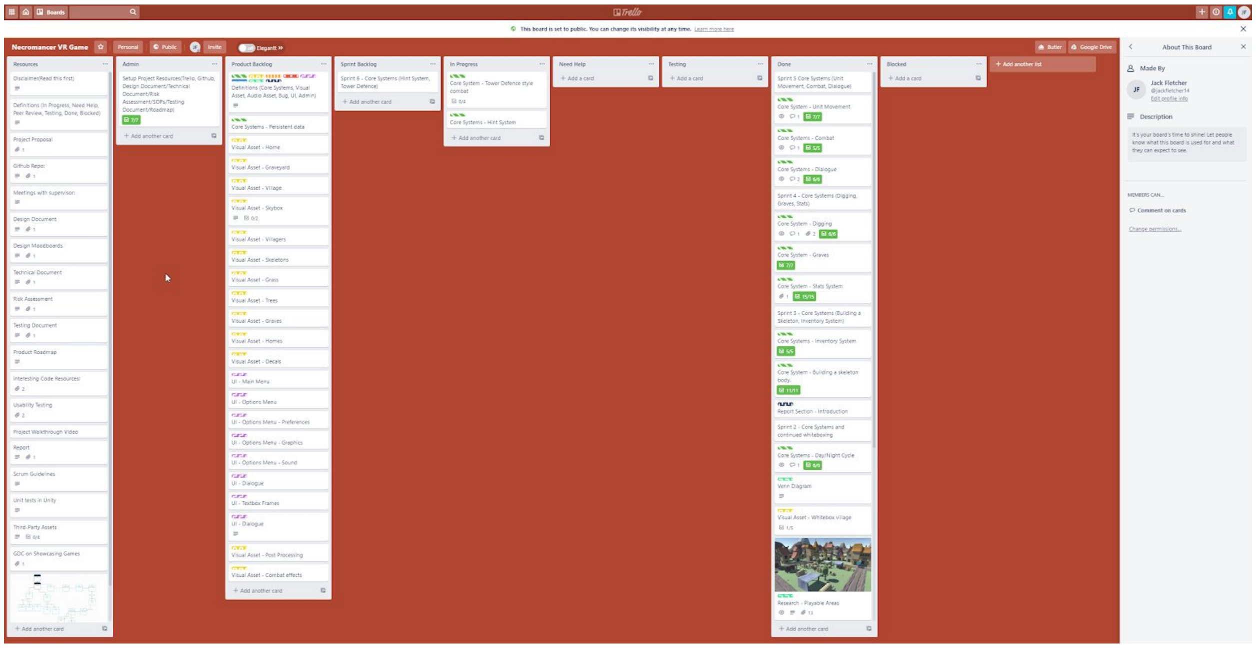
It's your board's time to shine! Let people know what this board is used for and what they can expect to see.

MEMBERS CAN...

Comment on cards

Change permissions...

Figure 20. Week 6 Trello submission.



Boards · Trello

This board is set to public. You can change its visibility at any time. Learn more here

**Necromancer VR Game** · Personal · Public · Invite · Elegantt · Butler · Google Drive

**Resources**
- Disclaimer(Read this first)
- Definitions (In Progress, Need Help, Peer Review, Testing, Done, Blocked)
- Project Proposal · 1
- Github Repo: · 1
- Meetings with supervisor:
- Design Document · 1
- Design Moodboards · 1
- Technical Document · 1
- Risk Assessment · 1
- Testing Document · 1
- Product Roadmap
- Interesting Code Resources: · 2
- Usability Testing · 2
- Project Walkthrough Video
- Report · 1
- Scrum Guidelines
- Unit tests in Unity
- Third-Party Assets · 0/4
- GDC on Showcasing Games · 1
- + Add another card

**Admin**
- Setup Project Resources(Trello, Github, Design Document/Technical Document/Risk Assessment/SOPs/Testing Document/Roadmap) · 7/7
- + Add another card

**Product Backlog**
- Definitions (Core Systems, Visual Asset, Audio Asset, Bug, UI, Admin)
- Core Systems - Persistent data
- Visual Asset - Home
- Visual Asset - Graveyard
- Visual Asset - Village
- Visual Asset - Skybox · 0/2
- Visual Asset - Villagers
- Visual Asset - Skeletons
- Visual Asset - Grass
- Visual Asset - Trees
- Visual Asset - Graves
- Visual Asset - Homes
- Visual Asset - Decals
- UI - Main Menu
- UI - Options Menu
- UI - Options Menu - Preferences
- UI - Options Menu - Graphics
- UI - Options Menu - Sound
- UI - Dialogue
- UI - Textbox Frames
- UI - Dialogue
- Visual Asset - Post Processing
- Visual Asset - Combat effects
- + Add another card

**Sprint Backlog**
- Sprint 6 - Core Systems (Hint System, Tower Defence)
- + Add another card

**In Progress**
- Core System - Tower Defence style combat · 0/4
- Core Systems - Hint System
- + Add another card

**Need Help**
- + Add a card

**Testing**
- + Add a card

**Done**
- Sprint 5 Core Systems (Unit Movement, Combat, Dialogue)
- Core System - Unit Movement · 1 · 7/7
- Core Systems - Combat · 1 · 5/5
- Core Systems - Dialogue · 2 · 6/6
- Sprint 4 - Core Systems (Digging, Graves, Stats)
- Core System - Digging · 1 · 2 · 6/6
- Core System - Graves · 7/7
- Core System - Stats System · 1 · 15/15
- Sprint 3 - Core Systems (Building a Skeleton, Inventory System)
- Core Systems - Inventory System · 5/5
- Core System - Building a skeleton body. · 11/11
- Report Section - Introduction
- Sprint 2 - Core Systems and continued whiteboxing
- Core Systems - Day/Night Cycle · 1 · 6/6
- Venn Diagram
- Visual Asset - Whitebox village · 1/5
- Research - Playable Areas · 13
- + Add another card

**Blocked**
- + Add a card

+ Add another list

**About This Board**

Made By

Jack Fletcher
@jackfletcher14
Edit profile info

Description

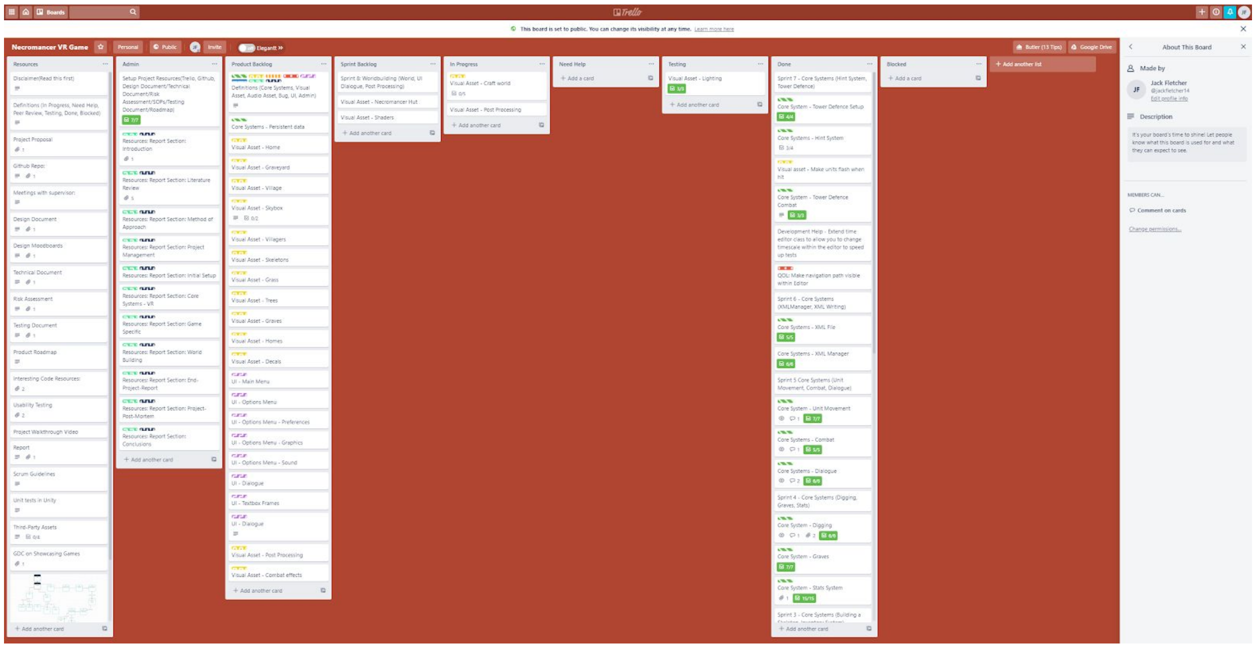
It's your board's time to shine! Let people know what this board is used for and what they can expect to see.

MEMBERS CAN...

Comment on cards

Change permissions...

Figure 21. Week 7 Trello submission.



Boards · Trello

This board is set to public. You can change its visibility at any time. Learn more here

**Necromancer VR Game** · Personal · Public · Invite · Elegantt · Butler (13 Tips) · Google Drive

**Resources**
- Disclaimer(Read this first)
- Definitions (In Progress, Need Help, Peer Review, Testing, Done, Blocked)
- Project Proposal · 1
- Github Repo: · 1
- Meetings with supervisor:
- Design Document · 1
- Design Moodboards · 1
- Technical Document · 1
- Risk Assessment · 1
- Testing Document · 1
- Product Roadmap
- Interesting Code Resources: · 2
- Usability Testing · 2
- Project Walkthrough Video
- Report · 1
- Scrum Guidelines
- Unit tests in Unity
- Third-Party Assets · 0/4
- GDC on Showcasing Games · 1
- + Add another card

**Admin**
- Setup Project Resources(Trello, Github, Design Document/Technical Document/Risk Assessment/SOPs/Testing Document/Roadmap) · 7/7
- Resources: Report Section: Introduction · 1
- Resources: Report Section: Literature Review · 5
- Resources: Report Section: Method of Approach
- Resources: Report Section: Project Management
- Resources: Report Section: Initial Setup
- Resources: Report Section: Core Systems - VR
- Resources: Report Section: Game Specific
- Resources: Report Section: World Building
- Resources: Report Section: End-Project-Report
- Resources: Report Section: Project-Post-Mortem
- Resources: Report Section: Conclusions
- + Add another card

**Product Backlog**
- Definitions (Core Systems, Visual Asset, Audio Asset, Bug, UI, Admin)
- Core Systems - Persistent data
- Visual Asset - Home
- Visual Asset - Graveyard
- Visual Asset - Village
- Visual Asset - Skybox · 0/2
- Visual Asset - Villagers
- Visual Asset - Skeletons
- Visual Asset - Grass
- Visual Asset - Trees
- Visual Asset - Graves
- Visual Asset - Homes
- Visual Asset - Decals
- UI - Main Menu
- UI - Options Menu
- UI - Options Menu - Preferences
- UI - Options Menu - Graphics
- UI - Options Menu - Sound
- UI - Dialogue
- UI - Textbox Frames
- UI - Dialogue
- Visual Asset - Post Processing
- Visual Asset - Combat effects
- + Add another card

**Sprint Backlog**
- Sprint 8: Worldbuilding (World, UI Dialogue, Post Processing)
- Visual Asset - Necromancer Hut
- Visual Asset - Shaders
- + Add another card

**In Progress**
- Visual Asset - Craft world · 0/5
- Visual Asset - Post Processing
- + Add another card

**Need Help**
- + Add a card

**Testing**
- Visual Asset - Lighting · 3/3
- + Add another card

**Done**
- Sprint 7 - Core Systems (Hint System, Tower Defence)
- Core System - Tower Defence Setup · 4/4
- Core Systems - Hint System · 3/4
- Visual asset - Make units flash when hit
- Core System - Tower Defence Combat · 3/3
- Development Help - Extend time editor class to allow you to change timescale within the editor to speed up tests
- QOL: Make navigation path visible within Editor
- Sprint 6 - Core Systems (XMLManager, XML Writing)
- Core Systems - XML File · 5/5
- Core Systems - XML Manager · 6/6
- Sprint 5 Core Systems (Unit Movement, Combat, Dialogue)
- Core System - Unit Movement · 1 · 7/7
- Core Systems - Combat · 1 · 5/5
- Core Systems - Dialogue · 2 · 6/6
- Sprint 4 - Core Systems (Digging, Graves, Stats)
- Core System - Digging · 1 · 2 · 6/6
- Core System - Graves · 7/7
- Core System - Stats System · 1 · 15/15
- Sprint 3 - Core Systems (Building a
- + Add another card

**Blocked**
- + Add a card

+ Add another list

**About This Board**

Made By

Jack Fletcher
@jackfletcher14
Edit profile info

Description

It's your board's time to shine! Let people know what this board is used for and what they can expect to see.

MEMBERS CAN...

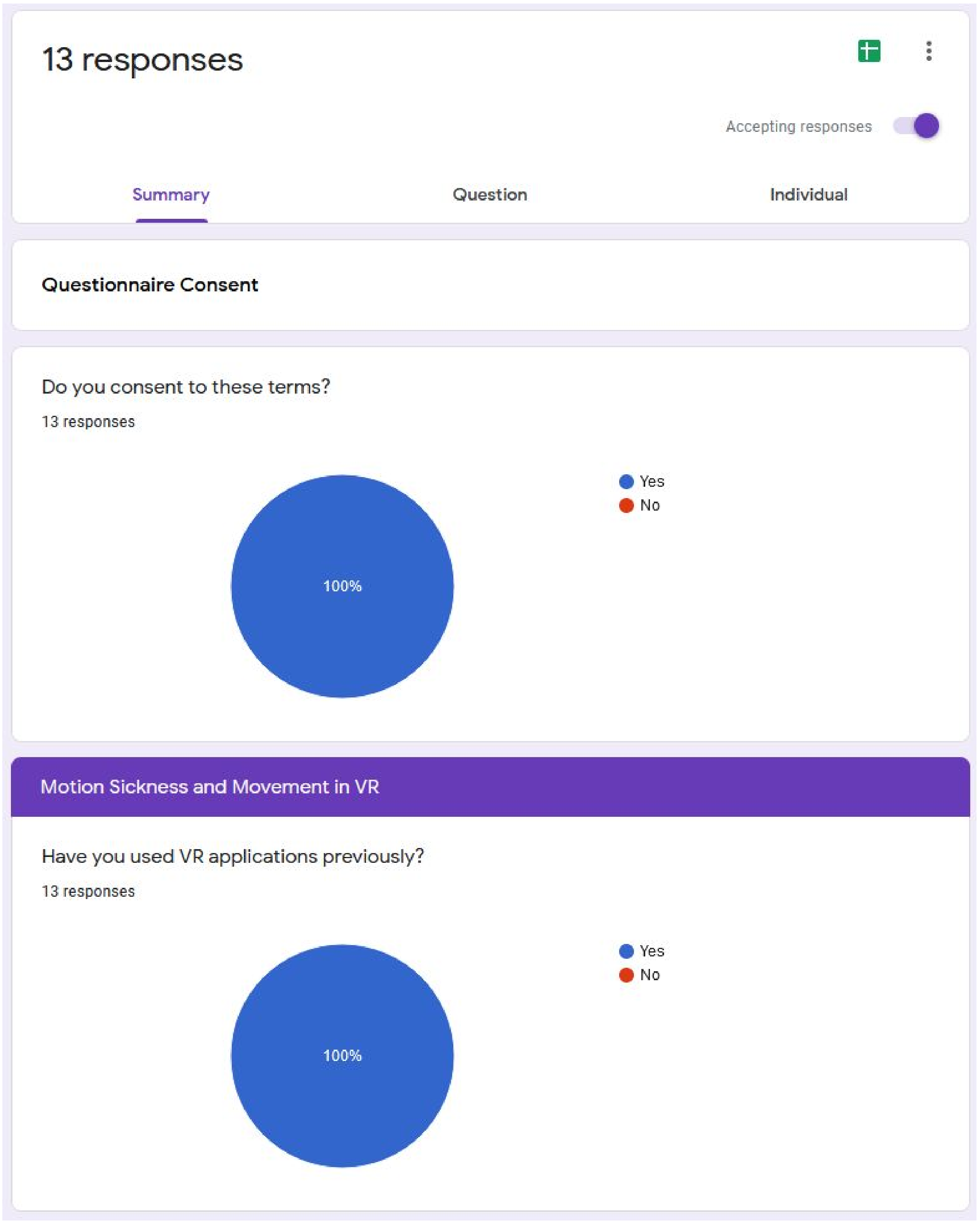
Comment on cards

Change permissions...

Figure 22. Week 8 Trello submission.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 2/5/2020 20:00:47 | Yes | Yes | PS4 VR | No | | Joystick | Ace Combat 7, No Man's | Yes | | |
| 2/5/2020 20:11:20 | Yes | Yes | Oculus Rift | No | | Real world space movement (Move your entire body | | Yes | | |
| 2/5/2020 20:23:19 | Yes | Yes | HTC Vive gen1 | Yes | Fast pace teleportation | Touchpad on handheld co | Windlands | Yes | | |
| 2/5/2020 20:41:22 | Yes | Yes | Google Cardboard, Oculu | Yes | games with lots of rotatin | Real world space moveme | Bones VR | Yes | | GL on your assignment |
| 2/5/2020 20:44:10 | Yes | Yes | Oculus Rift S | Yes | Clicky turning, i need con | Joystick | Vrchat and Pavlov | No | | |
| 2/5/2020 20:58:04 | Yes | Yes | Vive and index | Yes | Joystick controls or flying | Teleportation | The steam intro games | Yes | | |
| 2/5/2020 21:31:22 | Yes | Yes | Oculus rift S | Yes | The Anne Frank app, tele | Joystick | The walking dead vr | Yes | | Motion sickness is reduced when the right joystick allows the player to move the camera in increments rather than a smooth rotation |
| 2/5/2020 21:40:19 | Yes | Yes | HTC Vive, Index | No | | Teleportation | The Lab | Yes | | |
| 2/5/2020 22:21:51 | Yes | Yes | Oculus Rift | Yes | Pavlov VR Motion sick | Teleportation | Robo Recall | Yes | | |
| 2/5/2020 22:23:43 | Yes | Yes | Oculus Rift | Yes | Robo Recall and Beat Sa | Real world space moveme | No, haven't played that m | Yes | | |
| 2/6/2020 18:06:10 | Yes | Yes | Vive, Index, Cardboard | No | | Teleportation | Doom VR, valve labs thing | Yes | | |
| 2/7/2020 19:25:24 | Yes | Yes | HTC Vive (don't know whi | Yes | The game was first perso I suffer from car sickness | I couldn't pick a favourite; the only experiences I've t | | Yes | | |
| 2/10/2020 15:14:19 | Yes | Yes | HTC Vive, Oculus Rift | No | | Teleportation | | Yes | | |

*Figure 23.  Results of anonymous questionnaire as a spreadsheet*



13 responses

**Questionnaire Consent**

Do you consent to these terms?

13 responses

Motion Sickness and Movement in VR

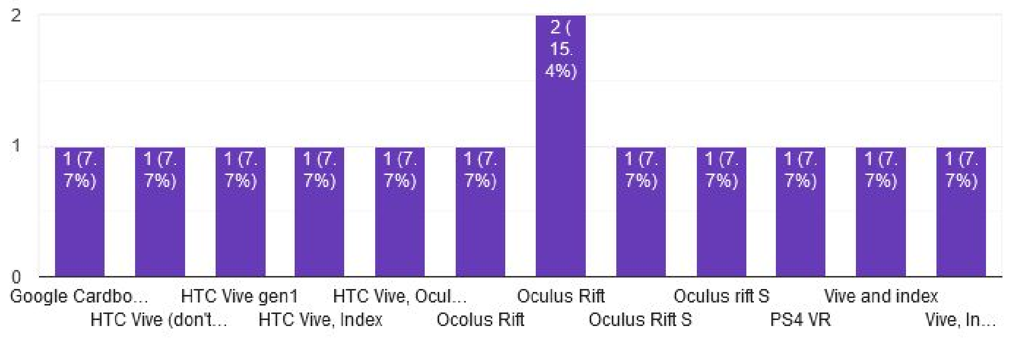
Have you used VR applications previously?

13 responses

*Figure 24. Questionnaire results page 1*

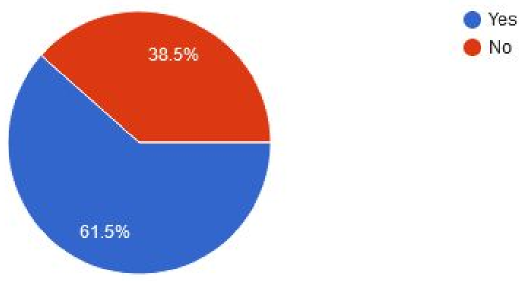
What VR equipment did you use for this experience? (E.G, Google Cardboard, Oculus Quest) If you're unsure, please write whether it was a mobile or desktop experience.

13 responses



Have you ever experienced motion sickness within VR Applications?

13 responses



If you answered yes to the previous question, can you give any further details? For example, the game that was played, the type of movement (E.G, drag to move, joystick controls, teleportation), a rough estimate of the time it took you to feel sick.

8 responses

games with lots of rotating and about 45min

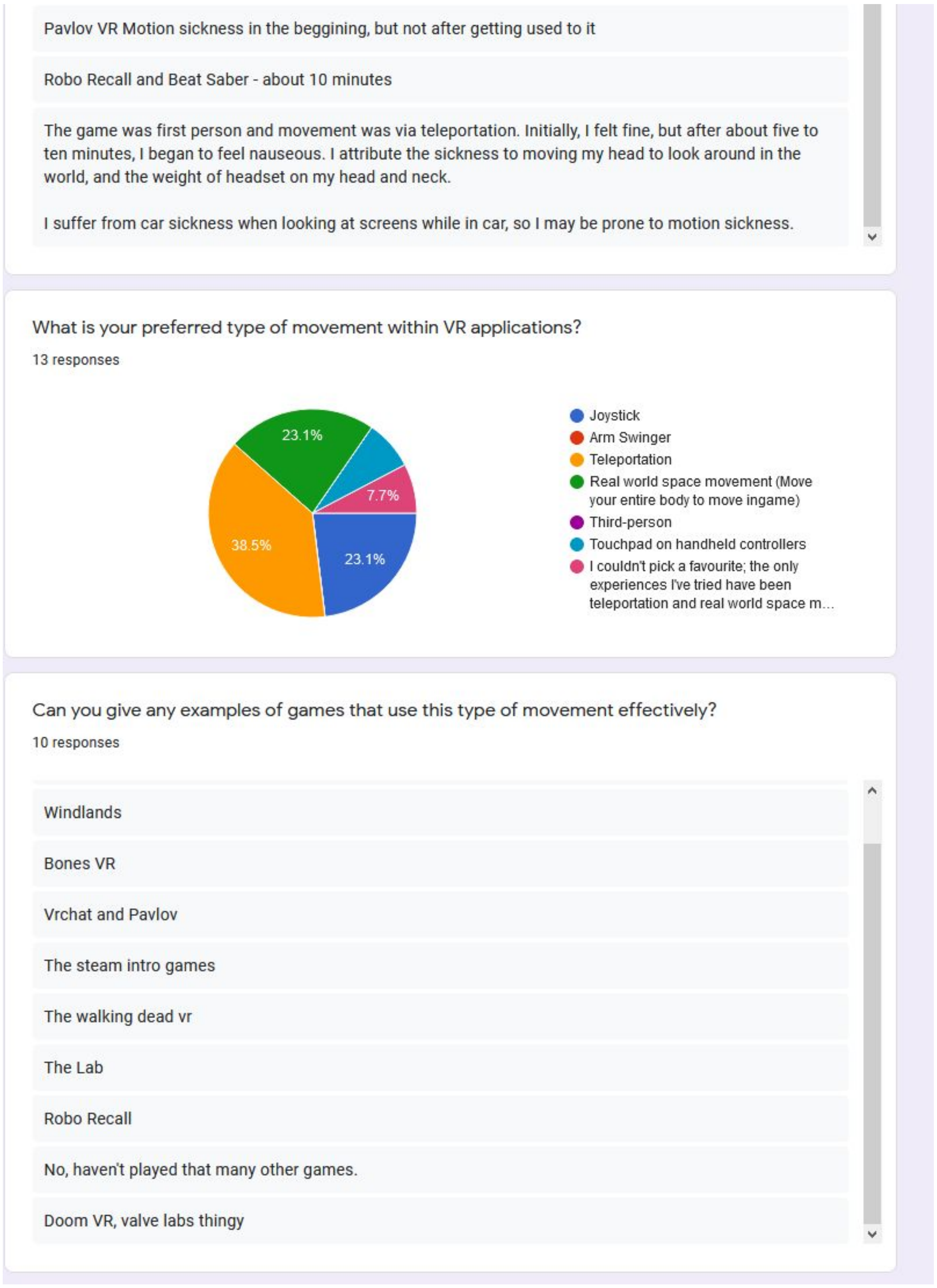
Clicky turning, i need comfort turning to feel well.

Joystick controls or flying make me the most sick and often I get sick pretty quickly

The Anne Frank app, teleportation, 5-10 minutes

Pavlov VR Motion sickness in the beggining, but not after getting used to it

*Figure 25.  Questionnaire results page 2*



Pavlov VR Motion sickness in the beggining, but not after getting used to it

Robo Recall and Beat Saber - about 10 minutes

The game was first person and movement was via teleportation. Initially, I felt fine, but after about five to ten minutes, I began to feel nauseous. I attribute the sickness to moving my head to look around in the world, and the weight of headset on my head and neck.

I suffer from car sickness when looking at screens while in car, so I may be prone to motion sickness.

What is your preferred type of movement within VR applications?

13 responses

23.1%
7.7%
38.5%
23.1%
● Joystick
● Arm Swinger
● Teleportation
● Real world space movement (Move your entire body to move ingame)
● Third-person
● Touchpad on handheld controllers
● I couldn't pick a favourite; the only experiences I've tried have been teleportation and real world space m...

Can you give any examples of games that use this type of movement effectively?

10 responses

Windlands

Bones VR

Vrchat and Pavlov

The steam intro games

The walking dead vr

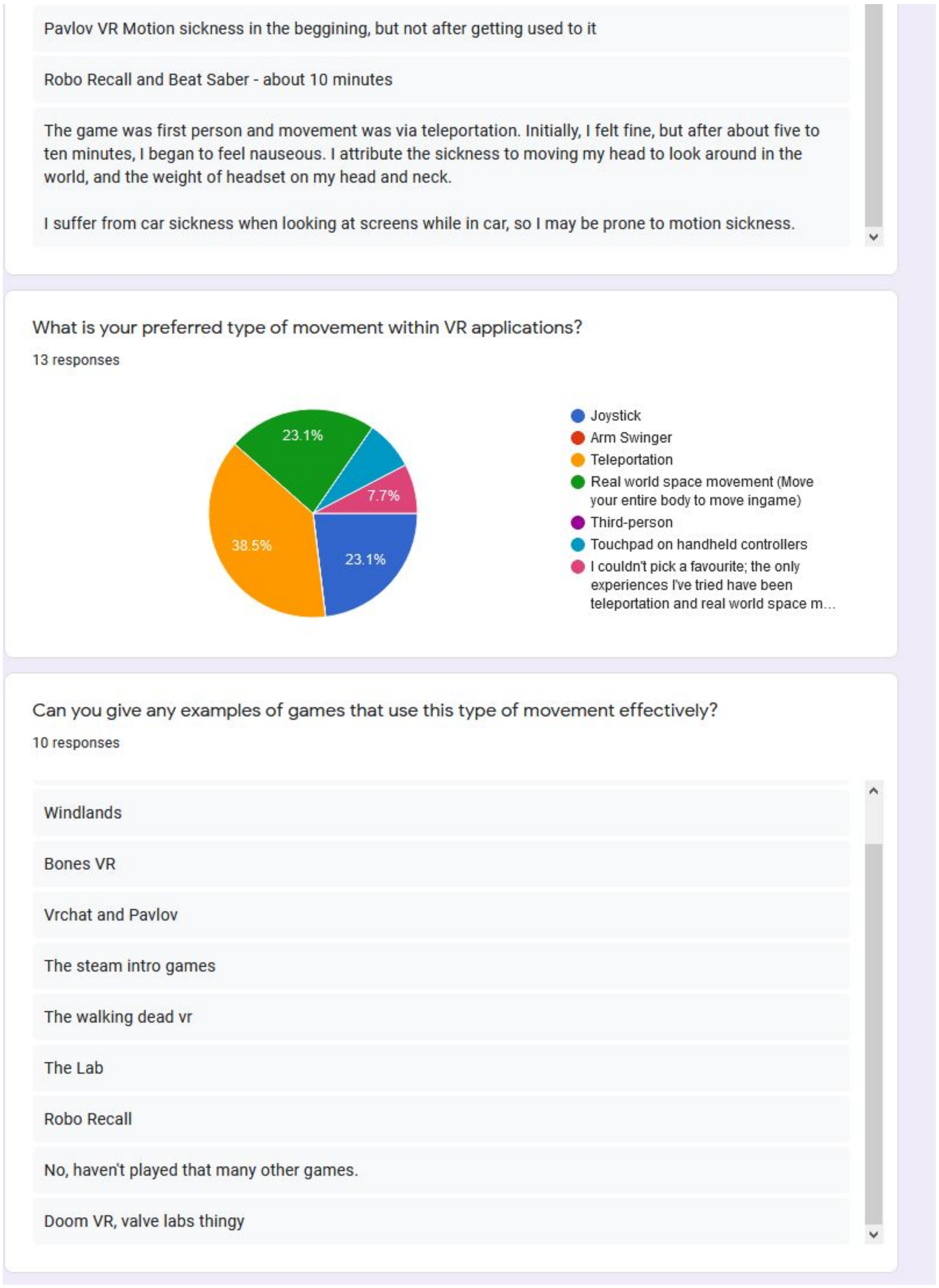
The Lab

Robo Recall

No, haven't played that many other games.

Doom VR, valve labs thingy

*Figure 26. Questionnaire page 3*


Pavlov VR Motion sickness in the beggining, but not after getting used to it

Robo Recall and Beat Saber - about 10 minutes

The game was first person and movement was via teleportation. Initially, I felt fine, but after about five to ten minutes, I began to feel nauseous. I attribute the sickness to moving my head to look around in the world, and the weight of headset on my head and neck.

I suffer from car sickness when looking at screens while in car, so I may be prone to motion sickness.

What is your preferred type of movement within VR applications?

13 responses

23.1%
7.7%
38.5%
23.1%
● Joystick
● Arm Swinger
● Teleportation
● Real world space movement (Move your entire body to move ingame)
● Third-person
● Touchpad on handheld controllers
● I couldn't pick a favourite; the only experiences I've tried have been teleportation and real world space m...

Can you give any examples of games that use this type of movement effectively?

10 responses

Windlands

Bones VR

Vrchat and Pavlov

The steam intro games

The walking dead vr

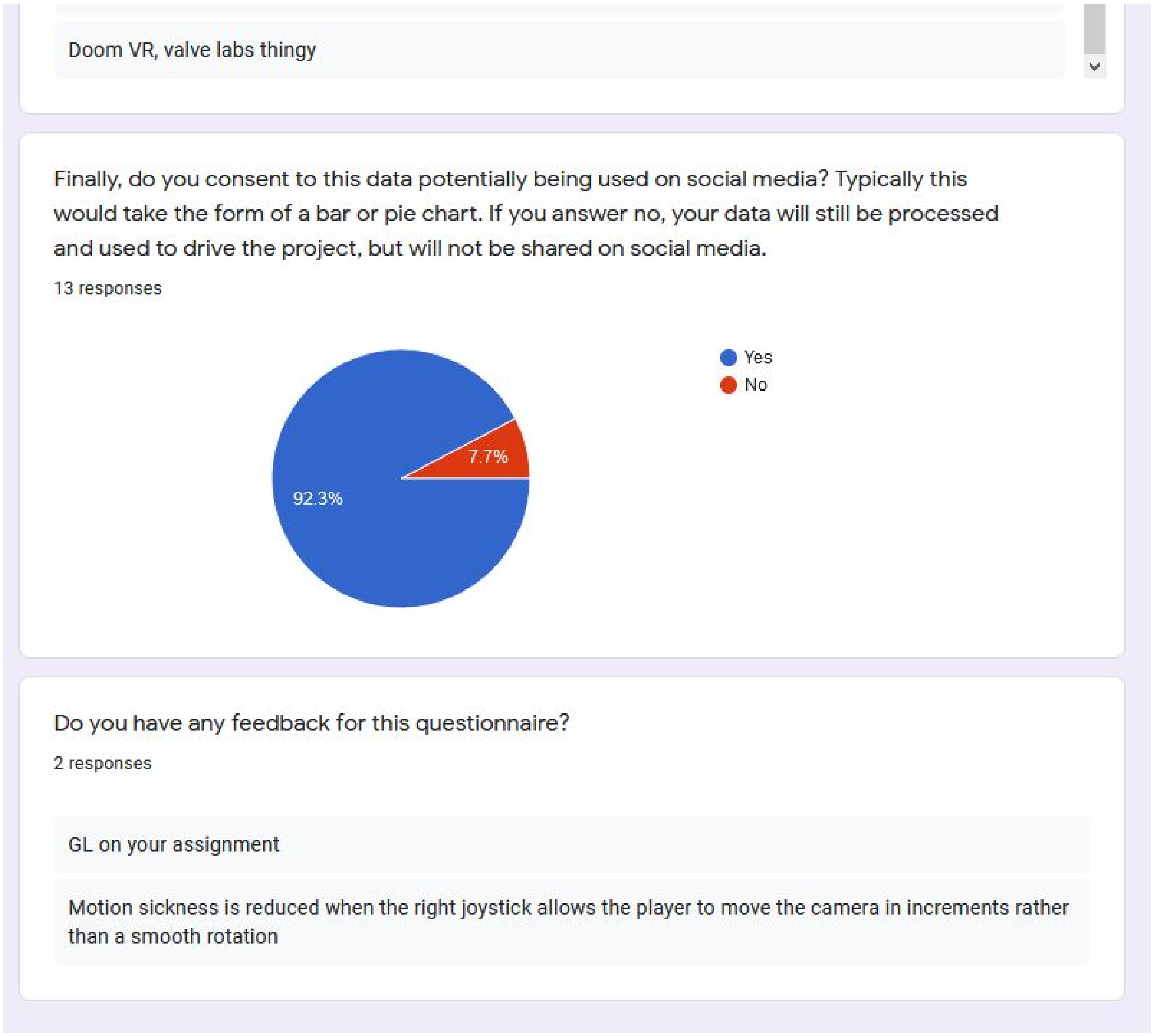
The Lab

Robo Recall

No, haven't played that many other games.

Doom VR, valve labs thingy

*Figure 27. Questionnaire results page 4*



Doom VR, valve labs thingy

Finally, do you consent to this data potentially being used on social media? Typically this would take the form of a bar or pie chart. If you answer no, your data will still be processed and used to drive the project, but will not be shared on social media.

13 responses

Do you have any feedback for this questionnaire?

2 responses

GL on your assignment

Motion sickness is reduced when the right joystick allows the player to move the camera in increments rather than a smooth rotation

*Figure 28. Questionnaire results page 5*

| Risk Category | Risk Explanation | Level |
|---|---|---|
| Schedule | Schedule is created with a best case scenario in mind, rather than being more realistic | High |
| | Sprints are not designed with other tasks in mind. I.E, Sprints are not filled with linked tasks that collectively make a function or a feature | Medium |
| | Issues that are not identified early in the project may impact meeting deadlines | Medium |
| | Incorrect prioritization of tasks leads to less crucial tasks being completed before core tasks, potentially stopping essential functions from being created with time constraints | Medium |
| | A similar game coming to market before this one can be completed | Low |
| Personnel | Personal strengths are misevaluated and time is required to adjust for this, affecting development time. | Medium |
| | Personal illness may occur and affect development time | Medium |

| | | |
|---|---|---|
| Design Process | Research in relevant areas requiring more time than expected. | Low |
| | User testing finds major issues with the design of the application - Requires major rework. | Low |
| | Inability to test on a variety of different VR devices, with various specifications. | Medium |
| | Overambitious scope, resulting in incomplete project | Medium |
| | Scope creep, where interesting game mechanics may take too long to implement to be viable | Medium |

*Table 3. Risk Assessment*

## 12.2. Links to Project Management:

| Project Management Tool | Link |
|---|---|
| Trello | https://trello.com/b/MWxqYi1Z/necromancer-vr-game |
| Technical Document | https://docs.google.com/document/d/1T-8ILRG5clrZCBRAFqzDYy7d9nVClDMHrn28Fd-tJrU/edit?usp=sharing |
| Risk Assessment | https://docs.google.com/document/d/1tEZR8Y821GhjwI7QWex3a0aRWHNk9UyCaNoY8cFhqcU/edit?usp=sharing |

| | |
|---|---|
| Testing Document | https://docs.google.com/spreadsheets/d/1VQU0AmNPiKPIxsObnOHRwWC339DUuX-boW0A-J3IybI/edit?usp=sharing |
| Design Document | https://docs.google.com/document/d/1a3FucSYpFFMVWPpao8jWg6WQxyPHBn1jkyev_gyUkXk/edit?usp=sharing |

*Table 3. Table of Project Management Links*

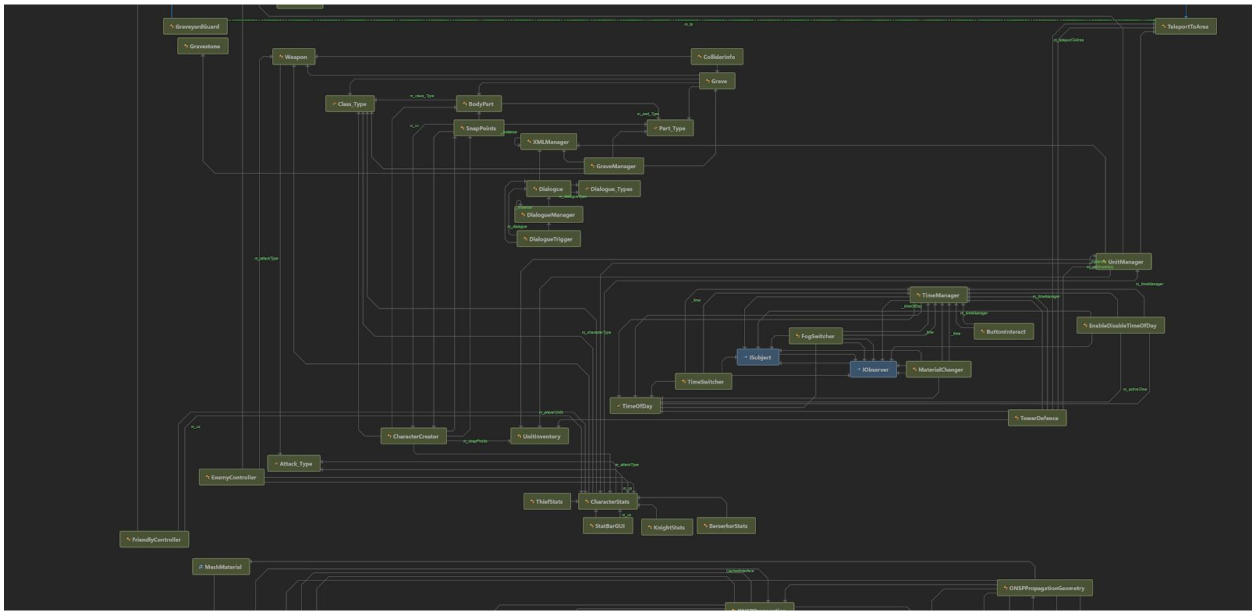


*Figure 29. Flow Diagram created by ReSharper (JetBrains, 2020)*