Bachelor's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

Leonard Sandløkk Schiller

Pac-Man in VR on the Omnideck

IDATT2900 Bachelor's project Computer Engineering

Bachelor's thesis in Computer Science
Supervisor: Alexander Holt
June 2023

NTNU
Norwegian University of
Science and Technology

Leonard Sandløkk Schiller

Pac-Man in VR on the Omnideck

IDATT2900 Bachelor's project Computer Engineering

Bachelor’s thesis in Computer Science
Supervisor: Alexander Holt
June 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Abstract

The classic problem in VR is achieving higher presence and immersion while avoiding the most common VR pitfalls. An issue remains regarding natural interaction in larger VEs. With fully natural interaction, the physical room size becomes a constraint. When using technology to solve this constraint, less than natural interaction becomes the new limitation. One attempt at solving this comes from Omnifinity in the form of the Omnideck. This is an omnidirectional treadmill that allows the user to walk in all directions to move in VR. This project is about recreating the retro game Pac-Man for the Omnideck in a VR environment. The main purposes are to serve as a demo game for the Omnideck, to demonstrate its capabilities, to be a starting point in researching its limitations, and to find design solutions that circumvent these limitations.

Sammendrag

Det klassiske problemet i VR er hvordan man kan oppnå høyere immersjon og tilstedeværelse for brukeren samtidig som man unngår de vanligste fallgruvene. Et gjenstående problem dreier seg om naturlig interaksjon i større virtuelle rom. Med helt naturlig interaksjon er det fysiske rommet ofte for lite. Med teknologi blir interaksjonen mindre enn naturlig. Et forsøk på å løse dette problemet kommer fra Omnifinity i form av Omnideck. Dette er en omnidireksjonell tredemølle som lar brukeren gå i alle retninger og brukes i samsvar med VR. Denne oppgaven går utpå å lage spillet Pac-Man i et VR miljø for å vise mulighetene med Omnidecket og som utgangspunkt for videre forskning for å undersøke hvilke begrensninger som oppstår og hvordan å motvirke dem.

Dedication

Thank you to my dear family for supporting me throughout my entire education and through the challenging times I experienced during the three years of my bachelor's course. Thanks to my friends Haakon Buljo and Theodor Holme for performing user tests and helping with the pictures used in this thesis. Thank you to Alexander Holt who helped me as my supervisor during this project.

12.06.2023 Trondheim Leonard Sandløkk Schiller

Expanded Task Description

This task was created by Alexander Holt for the Department of Computer Science at NTNU. The task description was to recreate a retro game such as Pac-Man or Frogger in a VR environment in combination with the omnidirectional treadmill Omnideck. In addition, the task had an implied goal to use the Omnideck as the main locomotion method to translate real-life walking into character movement in the game. However, the task description was short and intentionally vague to allow for creativity and leeway in the game development, the process, and the research.

These two games were suggested due to their main game mechanics and gameplay relying mostly on character movement, which makes them ideal use cases for the Omnideck. These games are also easily recognizable and simple games with a low barrier to entry for fresh players. In this task, I chose to make a game resembling Pac-Man. There were two important requirements for the finished game. It should resemble the original Pac-Man, transferred into a VR environment, and it should use the Omnideck for movement inside the game. These requirements were not specifically stated but were implied in the description and clarified throughout the development process. Several more requirements were extracted from the overarching task description. These are all described in the requirements document. (Requirements Document). In the document, several requirements describe how to make the game more closely resemble and behave like the original Pac-Man.

This project was primarily a development project with a broad and vague research goal of obtaining a deeper understanding of the Omnideck. This gave the creative freedom to choose a more specific research goal. The decided upon research goal was to find the limitations of the Omnideck, how they affected the user, and to find effective design solutions to mitigate the limitations effectively.

Table of Contents

Abstract i
Sammendrag ii
Dedication iii
Expanded Task Description iv
Table of Contents v
Figures x
Chapter 1: Introduction 1
1.1 Definitions 2
1.2 Interest 3
1.3 Problem Statement and Research Questions 4
1.4 Limitation 5
1.5 Scope and Constraints 5
1.6 Thesis Structure 6
Chapter 2: Theory and Previous Work 7
2.1 Essentials of a Video Game 7
2.2 VR 8
2.3 Why VR Games? 10
2.4 Former Technologies 11
2.5 The Omnideck 12
2.5.1 Description 12
2.5.2 Functionality 13
Chapter 3: Method 15
3.1 Introduction 15
3.2 Research Method 15
3.3 Technologies 17
3.3.1 Unity 17
3.3.2 Blender 18
3.3.3 Gitlab 19
3.3.4 HTC Vive, SteamVR, and Omnitrack Software 19
3.4 Development Methodologies 20
3.5 Development Method for this Project 21
Chapter 4: Results 23
4.1 Administrative Results 23

4.2 Development Results 24
4.3 Omnideck Findings and Research Results 29
4.3.1 Walking Straight 29
4.3.2 Turning 31
4.3.3 The Center 32
4.3.4 Speed Regulation 33
4.3.5 Head Turning 35
4.3.6 Tracking/Crawling 35
4.3.7 Terrain 36
4.3.8 Noise 36
Chapter 5: Discussion 38
5.1 Administrative 38
5.2 Development 39
5.3 Research 40
5.3.1 RQ1 41
5.3.2 RQ2 41
5.3.3 RQ3 43
5.4 Reflection 45
Chapter 6: Conclusion 47
6.1 Further Work 47
Societal Impact 48
References 50
Attachments 55
Pre-Project Plan A
Innholdsfortegnelse - 2 -
1. Mål og rammer - 3 -
1.1 Orientering. - 3 -
1.2 Problemstilling / prosjektbeskrivelse og resultatmål - 3 -
1.3 Mål - 3 -
1.4 Rammer - 4 -
2. Organisering - 5 -
3. Gjennomføring - 5 -
3.1. Hovedaktiviteter. - 5 -
3.1.1 Begrunnelse - 5 -

3.1.2 Forutsetninger - 6 -
3.1.3 Gjennomføring - 6 -
3.1.4 Resultater - 7 -
3.2. Milepæler. - 7 -
4. Oppfølging og kvalitetssikring - 7 -
4.1 Kvalitetssikring. - 7 -
4.2 Rapportering. - 8 -
5. Risikovurdering - 8 -
Project Handbook B
Innhold - 1 -
Framdriftsplan – Mileperlplan - 1 -
Møteinnkallinger - 2 -
Møteinnkalling 1 - 3 -
Møteinnkalling 2 - 4 -
Møteinnkalling 3 - 5 -
Møteinnkalling 4 - 6 -
Møteinnkalling 5 - 7 -
Møteinnkalling 6 - 8 -
Møteinnkalling 7 - 9 -
Møteinnkalling 8 - 10 -
Møteinnkalling 9 - 11 -
Møteinnkalling 10 - 12 -
Møtereferater - 13 -
Møtereferat 1 - 14 -
Møtereferat 2 - 16 -
Møtereferat 3 - 18 -
Møtereferat 4 - 20 -
Møtereferat 5 - 21 -
Møtereferat 6 - 22 -
Møtereferat 7 - 23 -
Møtereferat 8 - 25 -
Møtereferat 9 - 26 -
Møtereferat 10 - 27 -
Timelister - 28 -

Oppsummering av timelister - 29 -
Uke 4 - 30 -
Uke 9 - 31 -
Uke 10 - 32 -
Uke 11 - 33 -
Uke 12 - 34 -
Uke 13 - 35 -
Uke 14 - 36 -
Uke 15 - 37 -
Uke 16 - 38 -
Uke 17 - 39 -
Uke 18 - 40 -
Uke 19 - 41 -
Uke 20 - 42 -
Uke 21 - 43 -
Uke 22 - 44 -
Uke 23 - 45 -
Vision Document C
Innholdsfortegnelse - 1 -
Innledning - 2 -
Sammendrag problem og produkt - 2 -
Problemsammendrag - 2 -
Produktsammendrag - 2 -
Overordnet beskrivelse av interessenter og brukere - 2 -
Oppsummering interessenter - 2 -
Oppsummering brukere - 3 -
Brukermiljøet - 3 -
Sammendrag av brukernes behov - 3 -
Alternativer til vårt produkt - 3 -
Produktoversikt - 4 -
Produktets rolle i brukermiljøet - 4 -
Forutsetninger og avhengigheter - 4 -
Produktets funksjonelle egenskaper - 4 -
Ikke-funksjonelle egenskaper og andre krav - 4 -

Requirements Document D
Innholdsfortegnelse - 1 -
Introduksjon - 1 -
User Stories - 2 -
Spilloversikt - 5 -
Prototyper - 7 -
Wireframes - 7 -
Referanser - 9 -
System Document E
Necessary software: - 1 -
How to install: - 1 -
#1: Installing UnityHub and correct unity version - 1 -
#2 Installing Steam and SteamVR - 2 -
#3 Installing HTV Vive - 2 -
#4 Installing Omnifinity Omnitrack application - 3 -
#5 Downloading gitlab repo - 3 -
Opening application: - 3 -
Method 1 with executable: - 3 -
Method 2 in unity editor: - 4 -
How to play! - 4 -
Poster Presentation F
Gitlab G

Figures

Figure 1: The Omnideck at the Visualization Lab at NTNU 4
Figure 2: Top view of the Omnideck from the product spec sheet (Omnifinity, n.d.) 12
Figure 3: Flow of data between the used technologies 14
Figure 4: Research process for this thesis 17
Figure 5: Screenshot of the game in Unity Editor 18
Figure 6: Screenshot of cherry asset being created in Blender software 18
Figure 7: Screenshot of the git repo for this thesis on GitLab's website 19
Figure 8: HTC Vive Base station for tracking 20
Figure 9: HTC Vive HMD, Wireless adapter, Controllers and battery pack 20
Figure 10: Omnitrack proprietary software interface 20
Figure 11: HTC Vive wireless application 20
Figure 12: SteamVR application 20
Figure 13: Power Pellet (left) and regular Pellet (right) in the VR game 25
Figure 14: Screenshot of Pac-Man mobile game for reference Copyright of Bandai Namco . 25
Figure 16: One of the obtainable fruits in the VR game 26
Figure 15: Score displayed in the VR game 26
Figure 17: Multiple ghosts shown in the game along with a held item and the map 26
Figure 18: Teleportation wall in the game, walking through teleports the player to the other side of the maze 26
Figure 19: Map of the maze shows the player, ghosts, pellets, and items updated in real-time.. 27
Figure 20: A cherry bomb gun inspired by Pixel Gun 3D, drops from the cherry item 27
Figure 21: High score table displays after dying 27
Figure 22: Foot between rollers 29
Figure 23: Feet on separate segments 30
Figure 24: Top-down view of force vectors (red arrows) and foot placements (red circles) . 30
Figure 25: Foot on an exaggerated angle, F1 vector shows foot's natural backwards movement, F2 vector shows Omnideck segment's force vector 31
Figure 26: Feet sideways on the Omnideck while turning 32
Figure 27: One foot on the rollers, the other in the center 33
Figure 28: Omnitrack software, translation, and axes settings tab 33
Figure 29: Foot hits the Omnideck's outer edge 34
Figure 30: Looking around while walking forward 35
Figure 31: Crawling on the Omnideck 35
Figure 33: Sound level meter indicating 72.6 dB 36
Figure 32: Sound level meter indicating 47.2 dB 36

Chapter 1: Introduction

> "The ultimate display would, of course, be a room within which the computer can control the existence of matter. A chair displayed in such a room would be good enough to sit in. Handcuffs displayed in such a room would be confining, and a bullet displayed in such a room would be fatal." (Sutherland, 1965)

Since the dawn of VR, we have been progressing towards higher immersion and presence, to achieve as high a level of subjective realism as possible. The idea is that higher immersion and presence will provide the user with a higher-quality experience. (Schuemie, Van der Straaten, Krijn, & Van der Mast, 2001). A long-standing challenge with VR is bridging the gap between the computer-simulated VE and translating the experiences to a user who perceives them in the real world. Ideally, everything the computer simulates in the VE is experienced by the user as if it were subjectively real. Several immersive technologies have tried to close this gap between realities.

Modern HMDs feature high framerate displays with low latency, surround sound, tracking capabilities, and often come with controllers. They allow the user to perceive imagery, sounds, allow interaction in the VE, and contribute to higher immersion. In addition, the tracking capabilities of the HMD accurately translate the user's real-world movement within a predetermined confined space into in-game movement. This real-world movement is the best form of locomotion in VR both for presence, interaction, and prevention of cybersickness (Mayor, Raya, & Sanchez, 2019) and solves the problem of small-scale movement. However, many VEs are larger than the available area in the real world, so long-distance travel becomes a necessity. In these cases, real-world movement is no longer viable and other techniques are required.

In modern VR applications, larger-scale movement is often done either automatically, with no user input, or by the user's controller. Movement is often performed through either teleportation, where the user can either point to a new position using the controller and click a button to teleport to the new location, or through conventional button inputs, which move the user around in the VE.

Both leave more to be desired in the realm of presence and immersion compared to real-world natural locomotion. They can also worsen the problems of cybersickness from the disconnect of sensing movement in the VE but not in the real world (Mayor, Raya, & Sanchez, 2019).

To solve this, innovative technologies and implementations are needed for more realistic and immersive locomotion. As a result, multiple technologies have been presented over several years that attempt to keep the benefits of real-world walking without the constraints of a finite area. Some examples of these technologies include treadmills, step trackers, sliding surfaces, and even large “hamster balls”. (Nabiyouni, 2017). These are all different systems with distinct advantages and disadvantages, but the focus in this thesis will be primarily on the omnidirectional treadmill, the Omnideck, made by Omnifinity.

1.1 Definitions

API

Application programming interface, in this case the Omnifinity API used in Unity.

Blender

Open-source and free 3d modeling software.

Design science research

Pragmatic research method focused on development and creating prescriptive knowledge.

FDD

Feature-driven development, an agile development method with a focus on producing features.

Fidelity

Accuracy and exactness of reproducing desired effects.

Git

Distributed version control software.

HMD

Head-mounted display.

HTC VIVE

The HMD made by HTC.

Immersion

Immersion is an objective measurement of stimuli the user experiences. Full immersion is fully inclusiveness, extensive, surrounding, vivid, and matching.

Interaction

Object manipulation in both the physical and simulated world.

NTNU

Norwegian University of Science and Technology.

Omnifinity

Company that creates the Omnideck.

Presence

Describes the user's subjective feeling of being in another reality. Presence is increased by immersion. Akin to a “flow state”. (Csikszentmihalyi, 1975).

SteamVR

VR Software made by the Valve corporation.

Unity

Game development engine.

VE

The simulated virtual environment is a computer-generated reality which can be explored by the user.

VR

Virtual reality is an experience with a goal of fully immersing the user to create a sense of presence in an alternate simulated reality.

1.2 Interest

In December 2021 NTNU and the department of computer science purchased and installed the Omnifinity Omnideck in the Motion Capture and Visualization Laboratory at Gløshaugen Campus. The Omnideck is a flat, circular, omnidirectional treadmill with 16 segments of rollers that roll the user back toward the center when walking on them. (Chapter 2.5.1). Walking on this device while in the VE allows the user to experience movement in a novel way and provides the user with a new dimension of control to explore. Along with all original technologies, how to use it to its greatest potential is not yet fully understood. The goal of this technology is to provide higher quality and more immersive VR experiences. (Omnifinity, n.d.). Naturally, finding possibilities, limitations, use cases, and researching the model is valuable both for its use and for future researchers and developers who will use and test the system. As it is an expensive and novel piece of equipment, research on the Omnideck is uncommon and thereby further incentivized. Despite VR technology being widespread and commonly used for gaming, few if any VR games were developed with the Omnideck in mind. Though some conventional VR games can be played using the Omnideck for movement. (Aaru Entertainment, 2022). However, this still leaves a desire for a game specifically designed for the Omnideck, and which fully displays its capabilities.

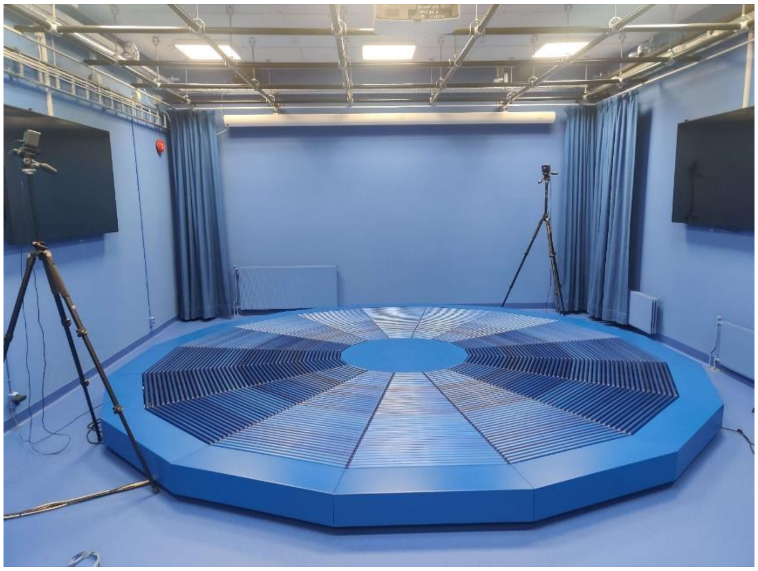


Figure 1: The Omnideck at the Visualization Lab at NTNU

With this background, I was tasked with the primary goal of making a game for the Omnideck which could demonstrate its capabilities, and which could be a starting point for further research and testing. A natural secondary goal of this research was to find further possibilities and limitations with the new equipment. (Pre-Project Plan). It is important to note that the project and research are purely for the betterment of future applications for the Omnideck and for researching the Omnideck. This project is not meant to reflect negatively on the products or brand of Omnifinity.

1.3 Problem Statement and Research Questions

Throughout the testing phase of the development process, multiple unstructured quality assurance tests were performed to test the game quality rather than the Omnideck's capabilities. However, on multiple occasions with multiple participants, similar difficulties were noticed and continued to reemerge. Some of these regarded normal walking, turning, and losing balance while walking on the Omnideck. These issues seemed to not be caused solely by user inexperience, as they often persisted despite the user's continued use of the system. The causes were unclear and suggested that innate limitations could be present that affect user experience.

Previous research has also been conducted on an adjacent topic regarding movement and interaction fidelity and how it influences user experience and VR locomotion. This research found that semi-natural locomotion techniques in VR had clear disadvantages and were outperformed by both higher interaction fidelity and high-quality lower interaction fidelity models. (Nabiyouni, 2017). This research further supported the theory of the existence of underlying limitations with the Omnideck that could cause abnormal experiences.

This spurred the research question for this thesis:

> "Which limitations and problems can arise and should be accounted for while developing applications in a VE with the Omnideck as a locomotion controller?"

The focus of this thesis will be on the specific limitations that were found with the Omnideck, how they affect the user, and how to account for or avoid them during the development of new applications. Three research questions were added to further flesh out the problem statement and provide research guidelines.

RQ 1:

What limitations exist with the Omnideck?

RQ2:

How can these limitations affect the user?

RQ3:

Can these limitations be counteracted?

1.4 Limitation

A limit, as defined by Merriam-Webster Dictionary is, "something that bounds, restrains, or confines". (Merriam-Webster, 2023). For this thesis, defining what is meant by a limitation is crucial as the focus is on the limitations of the Omnideck. The focus is primarily on how its design affects users' interaction with it, and what issues arise from this interaction. So, in this thesis, a limitation is defined as something that is not mechanically possible due to the Omnideck's design, or an interaction between the Omnideck and the user which causes or has the potential to cause unwanted effects during use.

1.5 Scope and Constraints

This thesis' scope is necessarily limited by the technology available, group size constraints, and time constraints.

The Visualization Lab at NTNU only houses the Omnideck and no other novel VR locomotion technologies. This confines the research to the Omnideck exclusively, since without testing possibilities, conclusive claims and comparisons cannot be made to other technologies. Other technologies fall outside of the scope of this project and are not researched in this thesis.

Time constraints limited the amount of research done in this thesis and were caused by two main factors. The group consists of one member, so less work could be done compared to a larger group. This project was meant to be for a group of two to three people, so as a single person, less work could be completed in comparison. The second factor was my participation in an exchange semester which only ended in early March, while the bachelor's project started in January. Both factors resulted in a heavy time constraint that affected the entire project. A less severe factor regards equipment sharing with other researchers during this study, since only one Omnideck is available. This time constraint meant that controlled experiments could not be performed. The lack of controlled experiments weakens the argumentation that the found flaws are significant, and more research is required to assert absolute severity.

1.6 Thesis Structure

The thesis' contents continue as follows:

- Chapter 1, Introduction to the contents of the thesis.
- Chapter 2, Theory and Previous Work covers important theory for games and VR that is necessary for the rest of the thesis.
- Chapter 3, Method covers the administrative, development, and research methods used in this thesis.
- Chapter 4, Results describe the administrative, development, and research results.
- Chapter 5, Discussion explains why the results were achieved.
- Chapter 6, Conclusion provides a conclusion for the research done.
- Societal Impact discusses the ethics behind the research topic.

Chapter 2: Theory and Previous Work

2.1 Essentials of a Video Game

Good game design has been a hotly debated topic where even the definition of game is undecided and has been a controversial topic in the past. (Stenros, 2016). There does not seem to be any cookie-cutter way to make a good game, and game-design rules are often broken by pioneering games. However, there is some agreement on core parts that are essential for a game to be a game. Chris Crawford, though having a more artistic and niche view on what makes a good game, has claimed that four essential parts of a game include "representation", "interaction", "conflict", and "safety". (Crawford, 1984).

Representation describes how a game is a "closed formal system that subjectively represents a subset of reality". (Crawford, 1984). In contrast to objective reality which, without getting too philosophical, is the real world and reality we live in, subjective reality is "a subset of our reality with its own rules, parts, and interactions". (Crawford, 1984). The game world is a "subjective reality". (Crawford, 1984). If the player is thoroughly immersed in and engaged with it, it will be convincing to the player. They will feel like the game world and the actions they take in it are real and of significance. An impressive game world with good lighting and sound effects are tells of a good game (Bond & Beale, 2009) and can contribute to making the subjective reality feel more real.

Interaction is also an essential feature of games. (Crawford, 1984). Without it, the user cannot play the game. A game where the player cannot interact with the game world and manipulate the environment could be comparable to a movie. High interaction quality is also one of the key components that make games good, with fast and functional feedback being among the most important features. (Bond & Beale, 2009).

Conflict is Crawford's fourth fundamental criterion for a game. (Crawford, 1984). When the player interacts with the game world to achieve a goal, conflict arises from obstacles or enemies that hinder the player. (Crawford, 1984). He claims that "games cleansed of conflict are not successful commercially and that few people enjoy them". (Crawford, 1984). Even in some modern games with seemingly no goals or conflicts, like sandbox games, players create goals for themselves, and conflict still arises when trying to achieve their self-made goals.

Lastly, Crawford states that safety is an important part of video games. (Crawford, 1984). While the conflict experienced in video games can feel emotionally real, at no point is the player in any objectively real danger. This means players can have dangerous experiences without the risks that are usually attached to the experience.

A final crucial factor of game design is making a game people want to play. There is no tangible way to determine a game's public reception before release, or how long the audience will stay if one ever forms, so making a game that people want to play is a nebulous task. Games and media are subject to the ebbs and flow of public opinion and trends so making a truly captivating game is difficult and unpredictable. This is a flaw Crawford himself acknowledged with his games, admitting there is some merit in saying his games "aren't that much fun". (Computer Gaming World, 1986). Despite the lack of a recipe for making a captivating game, it is possible to describe how the user feels while playing one. It creates feelings of concentration and captivation, along with a loss of awareness, surroundings, and a sense of time, while feeling rewarded by the challenges the game presents. This is like a flow state as described by Csikszentmihalyi, which he describes as "the optimal experience", which is desirable for a video game. (Csikszentmihalyi, 1975).

2.2 VR

The goal of VR is to allow the user to experience a new subjective reality as if it were real. The main difficulty is making the user feel fully immersed and thereby present in the simulated world. This difficulty comes from the innate problem of bridging the gap between the simulated and real world. Ideally, everything the computer simulates in the VE is experienced by the user as if it were subjectively real.

Immersion is an objective measurement of stimuli the user experiences, while an almost identical term, presence, instead describes the user's subjective feeling of being in the VE. A sense of presence is a psychological state that can be brought forth in a user to make them feel like they exist inside the VE. This is an illusion that needs to be upheld constantly to keep the user feeling present inside the VE.

Slater and Wilbur describe five key aspects to creating an immersive VE where the user can feel present, these are inclusiveness, extensiveness, surrounding, vivid, and matching. (Slater & Wilbur, 1997). The key aspects for an increased sense of presence are removing outside distractions, the extent of sensory information, the vividness and quality of the information, and the degree of interactivity and control the user has. (Schuemie, Van der Straaten, Krijn, & Van der Mast, 2001). These coincide with the key aspects of higher immersion. So, higher immersion can increase the likelihood and quality of the state of presence in the user.

Inclusiveness is defined as "the exclusion of the real world". (Slater & Wilbur, 1997). A fully inclusive VR would completely block out all elements of the real world and the user would only experience the virtual world. In a non-inclusive VE, the illusion of presence would be broken by stimuli from the real world. In some cases, this is desired, like with augmented reality, but for VR environments external stimuli should be avoided for maximum inclusion. Even seemingly banal reminders of the real world such as the external sound and the weight of an HMD can decrease the user's sense of presence. (Slater & Wilbur, 1997)

Extensiveness is defined as "the range of sensory modalities accommodated". (Slater & Wilbur, 1997). A fully extensive VE would accurately cover all human senses such as visual, auditory, olfactory, gustatory, haptic, tactile, vestibular, proprioception, thermoception, and nociception. Some senses, such as vision, are more important to stimulate than others. Vision contributes more to our perception than other senses since it is our primary source of information and takes precedence over other senses, (Kassuba, Klinge, Hölig, Röder, & Siebner, 2013) and around two-thirds of neural firings that happen in the brain per second come from the visual cortex. (Sells & Fixott, 1957). Therefore, stimulating our eyes is critical to creating an extensive VE, while stimulating other senses might be of lesser importance.

The VE must also fully surround the user. This increases their sense of presence in the environment. (Slater & Wilbur, 1997). Instead of having a narrow limiting view, for better immersion and presence, the VE must be panoramic. A larger field of view has a significant effect on the user's sense of presence. (Hendrix & Barfield, Presence within Virtual Environments as a Function of Visual Display Parameters, 1996) as referenced by (Slater & Wilbur, 1997). Instead of regular sound, spatialized sound or surround sound increases the sense of presence for the user. (Hendrix & Barfield, The Sense of Presence within Auditory Virtual Environments, 1996) as referenced by (Slater & Wilbur, 1997).

Another essential for an immersive VE is vividness, which regards the quality, richness, consistency, and fidelity of stimuli the user experiences. Vividness includes multiple types of fidelity. Within the domain of VR research, the term fidelity is divided into several categories.

The general meaning of fidelity can be described as "the quality or state of being faithful, or accuracy in details or exactness." (Merriam-Webster, 2023). This definition is accurate, but fidelity is often further divided into several types with specific names. Some examples are "representational", "experiential", (Jerald, 2015) "physical", "functional", "psychological" (Alexander, Brunyé, Weil, & Weil, 2005) "display", (McMahan, Bowman, Zielinski, & Brady, 2012) and "interaction" fidelity. (Nabiyouni, 2017), (Jerald, 2015) (McMahan, Bowman, Zielinski, & Brady, 2012). In this case, the most relevant definitions of fidelity are those describing the fidelity of user interaction with the environment, the fidelity of the user experiencing the environment, and the fidelity of the environment itself. These will be called interaction, psychological and physical fidelity, respectively.

Physical fidelity regards the objective quality and accuracy of the stimuli the system can provide the user. Some examples of physical fidelity include higher quality displays with higher framerates, lower latencies, higher resolutions, lower jitter, and more accurate colors. Lower fidelity in these aspects significantly decreases immersion and presence. (Louis, Troccaz, Rochet-Capellan, & Bérard, 2019). If physical fidelity is especially low-quality it can even negatively affect inclusion. (Slater & Wilbur, 1997). While higher physical fidelity increased presence, engagement, and usability for users. (McMahan, Bowman, Zielinski, & Brady, 2012).

Psychological fidelity regards users' subjective experience quality in the environment and is implicitly important for immersion and presence. If the user does not experience the VE as psychologically real, they will not feel present in the VE.

Interaction fidelity describes the objective quality and accuracy with which the system reproduces real-world interaction in the VE. This is crucial for upholding the illusion of interaction, which is essential for a high sense of presence in the user. Interaction fidelity is both how the user interacts physically in the real world with controllers and the resulting actions in the VE. It encompasses both locomotion and manipulation. A high degree of interaction fidelity means that the user can interact with the environment in many ways and that the interaction in the VE will closely resemble how the interaction would occur in the real world. Higher interaction fidelity is desirable since it leads to significantly increased presence, engagement, and usability. (McMahan, Bowman, Zielinski, & Brady, 2012). Interaction fidelity lies on a scale of naturalness ranging from completely natural to non-natural. (Nabiyouni, 2017). Natural interaction achieves the best results in presence, usability, and cybersickness, (Mayor, Raya, & Sanchez, 2019) but it has also been shown that lower fidelity super-natural interaction specifically with locomotion can have similar effectiveness to natural locomotion and superior effectiveness to semi-natural locomotion. (Nabiyouni, 2017).

Lastly, the user's experience inside the VE must be congruent with the user's real-world sensory stimuli. A mismatch between a user's sensory stimuli can cause reduced presence and increase the risk of cybersickness. An inverse relation exists between the two. (Weech, Kenny, & Barnett-Cowan, 2019). Cybersickness is an inherent and known problem within VR and can cause several symptoms such as disorientation, nausea, and headaches. (Davis, Nesbitt, & Nalivaiko, 2014). The most widely accepted theory behind cybersickness is the sensory conflict theory. This theory explains how incongruent senses cause sickness and nausea in people if multiple senses do not "agree" with one another. (Oman, 1990). An example of this could be standing in a room on a rocking boat, the vestibular system senses the rocking motion, while the visual system does not. In a VE, the cause of cybersickness could be from several smaller inconsistencies between senses, or larger sensory conflicts like using a controller to move, but not moving in the real world. Minimizing the mismatches of senses that occur from the differences between the real and virtual worlds is vital to both uphold presence and minimize cybersickness.

2.3 Why VR Games?

In many ways, video games and VR tackle the same problems and attempt to achieve similar goals. Combining the two would be mutually beneficial. Both games and VR strive for high user presence, which requires extensiveness, vividness, and interaction so the user's illusion of presence is upheld. Both have the overarching goal of providing new experiences without the risks or limitations of the real world.

Both need to be closed and inclusive systems that represent a new reality. In both cases, a new reality is formed for the user to be immersed in. Though this is more literal for VR and figurative for video games, both are similar challenges. Video games need good representation and would benefit from the higher level of immersion and presence that VR offers. The inclusiveness, extensiveness, and surrounding nature of a VE will help bring the player into the video game world. This contrasts with standard video games, where the player externally controls a character which could limit presence. Instead, in VR, the user is the character. The three-dimensional nature of the VE also provides the player with more gameplay possibilities compared to a two-dimensional flat screen.

In both cases, the system should be vivid. The environments in both VR and video games must be high quality, consistent, and rich. Inconsistencies, high latencies, low framerates, and bad visuals all have a negative impact on the overall quality of both video games and VR applications.

In both cases, the system should provide high-quality interaction. In VR, interaction is required both for higher presence and immersion. While for a video game, interaction is a requirement and high-quality interaction improves the game's quality. VR unlocks new forms of interaction that are not possible in conventional video games. Manipulation and locomotion in VR can be done physically in the world space instead of through pressing buttons on a keyboard. All these aspects can improve the interaction quality in video games.

Video gaming has also had a positive effect on the development of VR technologies as a whole. Video games are a mainstay in modern society. The industry has more than doubled in revenue from 2015 to 2021 from $95 billion to $191 billion worldwide and it is predicted to continue to grow. (Harding-Rolls, 2022). An excess of 3 billion people play video games worldwide. (Howarth, 2023). Video gaming is a massive industry and has already had a large positive influence on VR. The first HMDs, along with modern ones were marketed towards consumers to play VR video games. The first mainstream consumer-grade HMD was designed and marketed specifically to play video games. (Luckey, 2022) (Kedmey, 2015). This helped bring VR gaming to the public Zeitgeist, along with investors and interest from companies to create more VR technologies.

Despite the growing interest and many benefits of VR video games, they remain niche, with an estimated global user percentage of 1.3%. (Armstrong, 2023). The exorbitant requirements for VR equipment are the likely culprit. The extensive hardware requirements for both video games and VR mean that to have a worthwhile experience, the user would need a personal computer powerful enough to run the games, an HMD, controllers, and trackers to play in VR, all of which combined cost thousands of dollars, in addition to a room large enough play in. VR video games are a luxury that only the specially interested play.

2.4 Former Technologies

VR is not a new concept, and throughout history, several novel technologies were created to emphasize or show one or more aspects of what is now essential for VR.

One of the first of these novel technologies was the panorama. The first panorama-like paintings were discovered in Pompeii and dated back to around 20 A.D., which displays how long immersion has been sought after in the arts. (Grau, 2002). Robert Barker was the first to present a fully panoramic painting, of London, in London, in 1792. (Ellis, n.d.). The main feature was the 360-degree painting surrounding the audience which created a sense of immersion for viewers. Later in the 1830’s, stereopsis was first described by Charles Wheatstone (Wheatstone, 1838) and the earliest stereoscopes were created. Stereoscopes are devices for viewing separate images through each eye which together are perceived as three-dimensional images. Some consider this the earliest form of rudimentary VR. In 1929 the first flight simulator called the link trainer was created by Ed Link. (De Angelo, 2000). It was created with immersion and interaction in mind, as the goal was to accurately train new pilots. The device had a high degree of interactability, as the gauges and instruments responded to the pilot’s inputs. This device was highly effective and successful, with several hundreds of thousands of pilots in multiple countries trained using the simulator. The first HMD, the Telesphere Mask, was created in 1960 and the first true VR system, the Sensorama, was created in 1962, both by Morton Heilig. (USA Patent No. 3,050,870, 1962) (USA Patent No. 2,955,156, 1960). The Telesphere mask looks remarkably like modern-day HMDs and included three-dimensional moving images, in color, with peripheral vision, binaural sound, scents, and breezes according to the patent. The Sensorama was the first immersive multi-sensory VR technology and is often considered to be the birth of true VR. It was extensive and covered several senses such as visual, haptic, olfactory, and auditory. This was done through a color screen, fan, odor emitter, sound system, and a moving chair. It was made to expand the ways a viewer could watch a movie. Both were huge leaps in extensiveness and immersion for VR. In 1982 Thomas G. Zimmermann created the first hand-tracking device (USA Patent No. 4,542,291, 1982) which could track if the user's fingers were bent. This led the way for the Nintendo POWERGLOVE, created as an accessory for the NES years later in 1989, which tracked the hand movements of players to control the character on screen. (A.G.E Tech, 1989). Throughout the 1990’s multiple HMDs and VR machines were released. (Markoff, 1994) (Beep! Medadrive, 1994). In 2012 the first modern HMD was pre-launched on Kickstarter and raised over $2.4 million in donations. (Kickstarter, 2012). This significantly increased the hype around VR and several more high-quality products followed suit. One of these was the HTC VIVE HMD released in 2016 (Williams, 2016) which was used in this project. In the past 10 years, VR technology and popularity have reached an apex with several high-quality technologies and software being available for consumers.

Evidently, throughout VR history, constant progress has been made to increase user immersion and presence. In recent years with the emergence of the modern HMD, many of the problems facing physical fidelity have been solved to a great extent, leaving an old but new problem of increasing other forms of fidelity like interaction fidelity. Some of these technologies, including the Omnideck, try to tackle this problem.

2.5 The Omnideck

2.5.1 Description

The Omnideck is a 4.2-meter large, 0.2-meter tall, 1500kg hexadecagonal motorized omnidirectional treadmill produced by Omnifinity. It is divided into 16 equally sized wedge-shaped segments that feature motorized rotating metal rollers that act as a walking area. The inner circle in the middle of the Omnideck along with the outer skirt portion on each segment are stationary metal walking surfaces. (Omnifinity, n.d.).

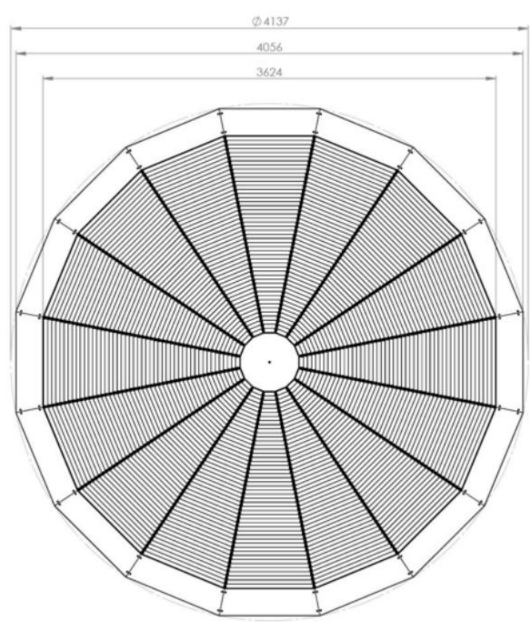


Figure 2: Top view of the Omnideck from the product spec sheet (Omnifinity, n.d.)

2.5.2 Functionality

The Omnideck's main purpose is to provide a 360-degree walking surface for VR applications. The key to understanding how the Omnideck works is in the design. There are two important parts of the Omnideck's design, the circular shape, and the roller segments. The circular shape is important as it allows the user to walk in any direction on the Omnideck. Starting from the center, no matter the direction the user walks in, they will be rolled back towards the center. This happens through the segments with rollers. The segments contain several rollers which roll towards the center when the user walks on them. This combination makes it so that no matter how or where the user walks, they will always be returned to the center of the Omnideck.

This rolling happens in combination with tracking. The Omnideck is used in combination with any HMD that features accurate tracking capabilities. When a user starts walking away from the center, the trackers on the HMD detect the position which is relayed to the Omnitrack application. This will then turn on the rollers in the segment that the user is standing on and roll the user back towards the center.

The Omnideck works "basically an analog joystick" (Thor & Omnifinity, 2023) that feeds movement vectors to a camera or character controller in a Unity application. It is analogous to a joystick both in the way movement is detected and sent to the game. Like a joystick, the further away an input is from the center, the larger the vector will be, and this vector is sent to the game to move the character. These vectors are created by the Omnitrack application which receives tracking data from the trackers on the HMD, then are sent to the Omnideck rollers to increase or decrease speed.

The rollers have different speeds depending on the distance the user is from the center of the Omnideck. The Omnideck features a smart speed control which increases or decreases the rolling speed depending on if the user is closer or farther away from the center. If the user is close to the center the rollers will move slowly, and when the user walks further towards the edge the rollers gradually increase the rolling speed until the user reaches the skirt of the Omnideck, where maximum speed is achieved. This speed modulation is entirely automatic and provides the user with a great deal of speed control.

The rollers are not part of the process that creates the vectors, so the Omnideck could theoretically function as a joystick independently of the rollers, though this is not intended and would entirely defeat its purpose. The Omnideck API is used for bi-directional communication between Omnitrack and the game. The movement vectors are sent to API in the game which then sends velocity vectors to the character controller which controls character movement. The velocity vector is applied to the in-game camera or character directly, so if this gameObject is misaligned with the calibrated play space, it will move in the wrong direction.

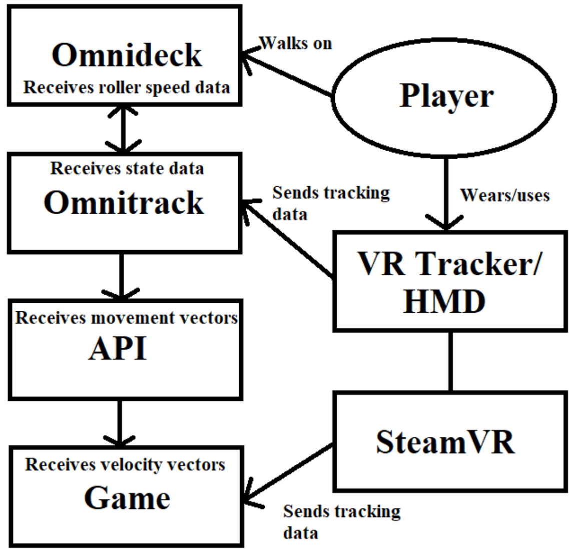


Figure 3: Flow of data between the used technologies

It is important to note that the in-game character will move both from the velocity added by the Omnideck API and the movement detected by the trackers on the HMD. These are separate movement types, the first being semi-natural and the second being fully natural. The movement from the tracked HMD is constantly applied to the in-game character, while the movement from the Omnideck is only applied when the user walks on the rolling segments. The movement added from the Omnideck moves the character larger distances by adding a velocity to the character controller based on the player's position on the Omnideck.

The HTC Vive has trackers that can send data to the HTC Vive application. The Vive software sends data and works with SteamVR and the Omnifinity application. SteamVR and the Omnifinity application send tracking data to the player's character controller so that when the user walks on the Omnideck, the character moves in-game and when the player moves their arms or head, the same actions happen to the in-game hands and viewport. These applications all work together to translate the movement of the user to the in-game character.

Chapter 3: Method

3.1 Introduction

At the base of scientific research lies the scientific method. The scientific method as defined by Merriam-Webster Dictionary is

> "Principles and procedures for the systematic pursuit of knowledge involving the recognition and formulation of a problem, the collection of data through observation and experiment, and the formulation and testing of hypotheses". (Merriam-Webster Dictionary, 2023).

It is important to follow these principles and procedures to assure the research and results are unbiased, true, and ethically and correctly obtained. The scientific method begins by obtaining knowledge within a domain and finding a question or observation that is worth researching further. From the observation or question a more thorough and testable hypothesis can be formed. The hypothesis could be a problem statement and include multiple underlying research questions. There are multiple research methods and strategies which can be used to collect data that can help answer the research questions. Some of these research methods include conducting surveys, experiments, case studies, or finding observations. These produce either qualitative or quantitative data types and the difference is significant. Quantitative data deals with numbers, can be statistically analyzed, and requires a larger amount of data points, while qualitative data deals mostly with words and descriptions, needs fewer data points, but is harder to reproduce, and can have a higher risk of bias. After producing data with research methods, the data must be analyzed, and a conclusion is formed to either confirm or refute the hypothesis, with empirical data or knowledge as the end product. Lastly, iteration is of high importance for scientific research, as hypotheses and theories are constantly tested for validity, and mistakes can be made in previous research.

The scientific method is separate from development techniques. Instead of helping to produce knowledge, development techniques assist developers with producing finished products.

3.2 Research Method

This thesis began as a wish to obtain a deeper understanding of the possibilities and limitations of the Omnideck, while creating a game designed specifically for it. The scientific research goal was to find limitations present with the Omnideck due to its design or use, gauge how those limitations could affect users or gameplay, then find ways to circumvent or mitigate the limitations through specific design choices. In addition to these motivations, earlier research on similar technologies serves as experience. Other technologies that produce semi-natural locomotion have been found to have many limitations and perform worse than natural or supernatural forms of locomotion. (Nabiyouni, 2017). This experience further motivates interest in research of the Omnideck. These initial motivations colored the research of this thesis and the design of the game.

These motivations were used to create a problem statement and the three distinct research questions: What limitations exist, how can they affect the user, and how can they be counteracted? These research questions were used to guide the research and divide the problem statement into distinct manageable parts. Most importantly, the questions can be used to provide conclusions for the research goal.

The primary research strategy used throughout the entire development and research process was design science research. Design science research is a more pragmatic scientific research strategy often used in computer science, research, and development, to develop knowledge used to design solutions to relevant problems. (Vaishnavi & Kuechler, 2004). This methodology has a focus on using the knowledge gained to evaluate and iterate artifacts to further improve performance. The strategy also focuses on the decision-making process, and how to make good or correct design decisions. This is inherently different from conventional science which has a goal of descriptively explaining how things are, while design science research is prescriptive and declares how things should be.

This divergence from the natural sciences comes with innate advantages and disadvantages. This research methodology is well suited for the given topic and research questions since both are about designing solutions to relevant problems in VR. The Omnideck offers a solution to the problem of locomotion in VR, while the research questions are used to evaluate, then suggest further design solutions to improve performance. The research strategy works well because there exists a prescribed theoretical optimal solution that any proposed solution can be compared to, then iteratively improved. This also provides substantial knowledge that guides the decision-making process for future developers and researchers who continue research within the domain.

This advantage is also the downside of this research strategy. The moment a prescriptive goal is held, the research strategy becomes biased, in contrast to natural science which has a goal of objectively describing how things are. When research is directed towards a prescribed goal, it can introduce conflicts of interest, or bad objectives that drive counterproductive research.

In combination with a design science research strategy approach, two data generation methods were used, observation and evaluation. These methods were relevant to both the research strategy and the research questions. Careful observation was useful for finding limitations, the cause behind these limitations, what effect they had on the user, and was used in combination with evaluation when evaluating potential solutions. Evaluation was useful for evaluating the problems that the user experienced and to consider potential solutions. Both methods were used during the user tests. They were used to test the game, the Omnideck, and potential solution implementations. Observations were often useful to answer research questions and were noted down throughout the entire development process. Whenever a test was performed, there was a specific focus on making meaningful observations, they were then written down during or directly after the test. After a test, the combined observations were evaluated in the context of the system to influence further design choices and to answer research questions.

There were multiple benefits to these methods. These included the high relevancy to the research strategy, the faster research speed, the low time requirements, and the fact that these methods can be performed iteratively. There are also risks of these methods producing biased or incomplete information that is difficult to reproduce.

Qualitative data was generated through these methods and was analyzed before iterating the process and improving design solutions. The qualitative data was divided by observations and evaluations and was relevant to either the research questions or improving the design.

Qualitative is distinct from and has innate advantages and disadvantages compared to quantitative data. Qualitative data is textual and descriptive in nature and is more subjective compared to objective numerical data. The advantages of this data type are that it can be used to directly describe causes and assists in gaining insight to find issues. This is highly relevant to the problem statement as the goal is to find limitations and design solutions. The disadvantages of qualitative data are the validity, reliability, and replicability of the data due to its subjective nature.

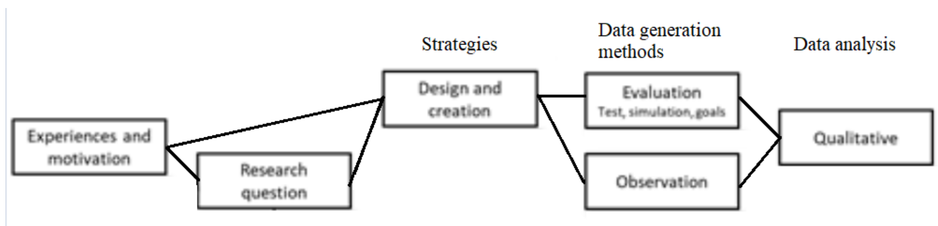


Figure 4: Research process for this thesis

3.3 Technologies

Throughout the development process, several important decisions were made regarding the choice of technology used, design choices, and work methodology. All these choices were made in the context of being a solo developer throughout the entire project, which put substantial importance on maximizing speed and productivity.

3.3.1 Unity

Unity is one of many game development engines and was chosen for this project for its distinct features that were beneficial for the quick development of the game. The first important reason Unity was chosen as the development game engine was prior experience and use of the software. The prior experience with Unity meant that there was less starting lag compared to if another game development engine was chosen. The second reason unity was chosen is how commonly it is used. Unity is by far the most used game engine for game development (Program-Ace, 2023) which means lots of resources exist online that help speed up the development process. Thirdly, Unity scripting natively uses the C# programming language. This language is similar to Java, which has been taught extensively throughout my bachelor studies. C# is comparatively faster and easier to write code for compared to other game engine scripting languages like C++ which is used in Unreal

Engine. There is a plugin to write scripts for Unreal Engine in Java, however, this is not native, and could potentially take more time. Unreal Engine also has the possibility of using nodes instead of scripts which is ideal for smaller projects and development speed, but due to unfamiliarity, this could have taken longer to learn. Lastly, other game development engines were not considered since Omnifinity, as of writing this, only has APIs for Unreal Engine and Unity.

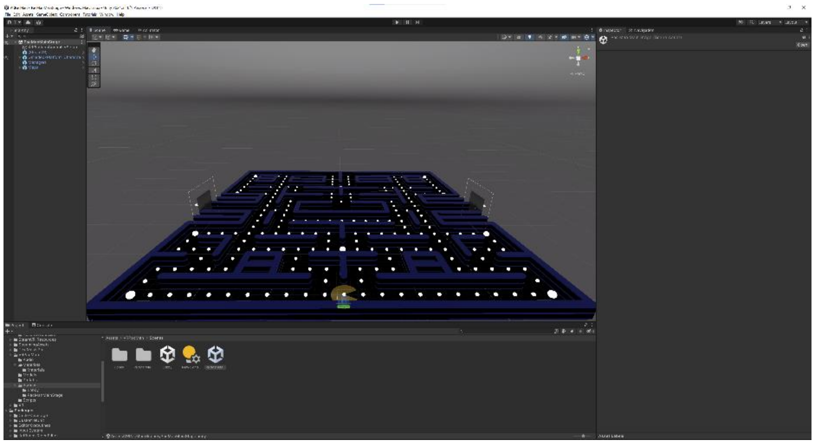


Figure 5: Screenshot of the game in Unity Editor

3.3.2 Blender

All assets were created using the 3d modeling software Blender. This software was chosen due to it being open-source, license-free, and most importantly, free software. These characteristics also fall in line with what NTNU prioritizes and values. Using 3d modeling software was necessary to make certain assets that did not exist in the vast unity asset store and was also the best option to avoid licensing difficulties. Despite having no previous experience with the software, it did not take too much time to understand and start making assets. Lastly, the created models were easily exported as FBX files and could easily be imported into Unity as prefabs. Blender has good integration with Unity and blend files can even be imported directly into Unity as prefabs.

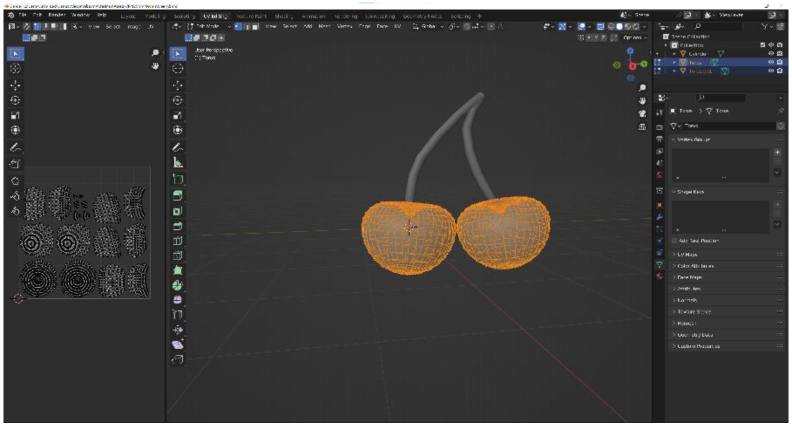


Figure 6: Screenshot of the cherry asset being created in Blender software

3.3.3 Gitlab

A version control system was necessary for this project and Gitlab was "chosen" as it was the only allowed version control system according to the requirements. Gitlab was used through NTNU's self-hosted GitLab for the Department of Computer Science.

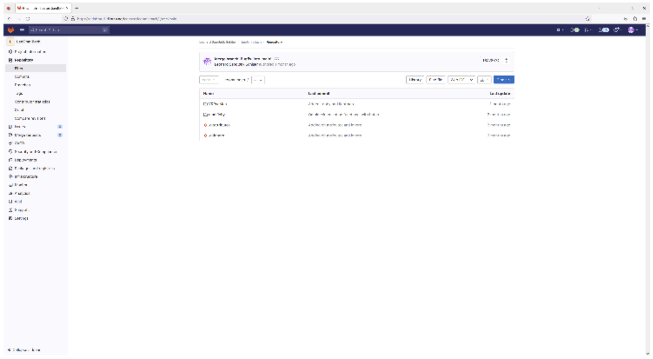


Figure 7: Screenshot of the Git repo for this thesis on GitLab's website

3.3.4 HTC Vive, SteamVR, and Omnitrack Software

The HMD used was the HTC Vive with a wireless adaptor in combination with SteamVR and the Omnitrack software. The HTC VIVE was the HMD that was available at NTNU's Visualization Lab, and the VIVE software works innately with the other software chosen and was therefore chosen. The SteamVR application and the SteamVR asset pack inside of Unity was used, as it worked with Unity and the Omnideck API. It also contained multiple valuable assets that were used for interaction, such as the hands, laser pointer, actions, and interactable script. The Omnitrack software was required for the use of the Omnideck.

Figure 9: HTC Vive HMD, Wireless adapter, Controllers, and battery pack

Figure 8: HTC Vive Base station for tracking

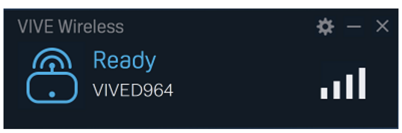


Figure 11: Screenshot of HTC Vive wireless application

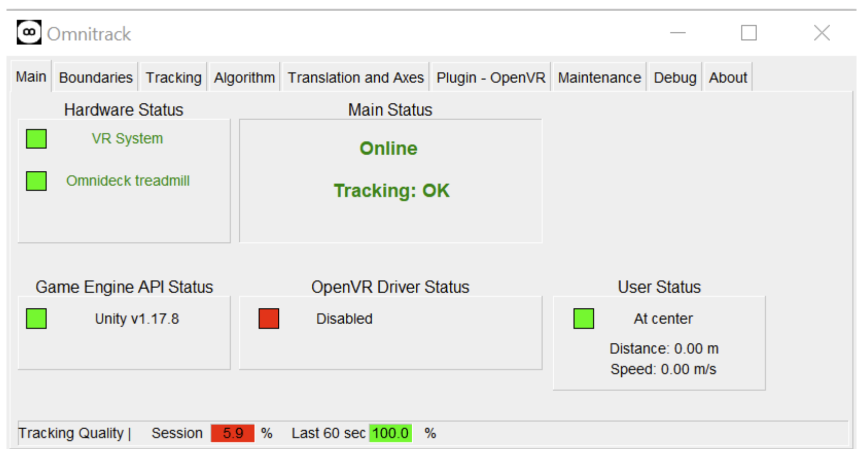


Figure 10: Screenshot of Omnitrack proprietary software user interface

Figure 12: Screenshot of SteamVR application

3.4 Development Methodologies

Multiple development methodologies exist and are used in different contexts for different goals. Two distinct types of development methodologies are Agile and Waterfall. Waterfall methodologies take a sequential approach to software development, starting with understanding requirements, creating a design, implementing the design, testing the product then delivering the finished system. This design process is straightforward but is rigid, does not allow for a change in requirements, and does not deliver the product fast. The second type of development method is called Agile. There are multiple types of agile software development methodologies, but they all follow general principles such as the importance of customer satisfaction, software delivery early in the process, iterative and flexible development to allow changing of requirements, frequent software delivery, and cooperation between developers and the product owner. (Beck, et al., 2001). These overarching goals allow the development process to be more iterative, flexible, and produce products more in line with the product owner's requirements. Agile methodology also uses many agile practices to help achieve the overarching goals. Some examples include user stories to describe product requirements, refactoring code for better structure, and having scrum meetings in larger teams.

3.5 Development Method for this Project

Development methodologies are usually tailored to groups, so following one of these methodologies precisely as a solo developer would lead to less productivity and longer development times, which is exactly what the methodologies are made to prevent. Due to this innate limitation, no methodology was chosen to be followed to a tee, however, practices from these methodologies were used and the development process was similar to the agile feature-driven development (FDD) method. FDD is lightweight since it has few rules and primarily focuses on quickly delivering features to the product owner. The method has 5 steps, the first two steps are sequential and the last three are iterative. Initially, in FDD, a system model is developed to understand the system's scope and requirements. Afterward, a list of required features based on the model is created. Then an iterative process of planning, designing, and then producing each feature is started.

For this project, the FDD approach was imitated due to its lightweight nature, lessened focus on teamwork, and heightened focus on adding features. Initially, an understanding of the scope and requirements for the game was created. This can be seen in the early versions of the vision and requirement documentation. (Vision Document) (Requirements Document). During this phase, the focus was on understanding the wishes of the product owner and accurately noting down the required features for the game. After the planning phase, the iterative process of adding features began. Multiple features were added and shown to the product owner every sprint, who gave feedback and insight on current and future features. Occasionally new features were requested by the product owner and were added to the requirements documentation. This agile method of development was beneficial to the project, as it increased flexibility, speed, and productivity of coding. The feature-driven approach allowed more features to be added to the finished version of the game due to the increased communication and validation from the product owner freeing up more time.

During this project, several agile processes were utilized which improved the development process overall. Initially, user stories were created to get a better understanding of requirements and necessary features. Agile testing was utilized by constantly testing the game for bugs and undesired events, this led to an increased game quality both during product owner meetings and for the end product. Lastly, Code refactoring was performed to increase code quality and ease the addition of new features.

Some common agile processes were completely avoided due to the increased workload and negligible benefit they would provide, or the impossibility of completing these processes alone. Some examples of avoided agile processes are scrum meetings, pair programming, and timeboxing.

Chapter 4: Results

4.1 Administrative Results

This project had multiple administrative goals with several tasks, deliverables, documents, and processes to be completed. Many items are a combination of the four.

The first deliverables were the pre-project plan and the vision document. (Pre-Project Plan). The pre-project plan describes the outline of the project, problem statement and research goals, effect goals, scope, organization, carrying through, follow-up, quality assurance, risk assessment, and time plan for the project. The vision document describes the problem and product, stakeholders and users, user environment, and a product description. (Vision Document). The pre-project plan served as a starting point and guide for the rest of the project while the vision document described the goal and vision of the overall system to be built. In total, around 26 hours were spent on these documents.

The second deliverable was the poster and its presentation. (Poster Presentation) The poster was a summary of the project and problem statement which described the case and work method. This was then presented to other bachelor students to discuss the project and receive feedback. Both were delivered and done within the deadline. In total, around 13 hours were spent on these deliverables.

The third deliverable was the project handbook. (Project Handbook). From the start of the project, multiple documents and processes had to be fulfilled related to the project handbook. This is a deliverable containing a project time plan, meeting notices, minutes, and status reports with working hour lists. The project time plan was a milestone plan showing the most important dates throughout the project and the deadlines of individual deliverables. In total, around 20 hours were spent on these deliverables.

Throughout the entire project, meetings were held between the advisor and group member. A meeting notice was sent in advance of every meeting detailing the duration and contents of the meeting as required, for increased time efficiency and as meeting preparation. Every meeting notice was saved as required. During the meeting, previous meeting minutes and status reports were discussed. After every meeting, minutes were written down, detailing information from the meeting for every important item on the agenda. These were utilized for the development, research, and writing of this project. All meeting minutes were saved as required. Lastly, status reports were updated often, detailing hours spent working on different tasks every week. This was meant to both log that enough working hours were completed, to show the progress throughout the project, and to show at what times different tasks were worked on. In total, around 8 hours were spent on meetings.

Lastly, and most importantly, the final deliverable is the main thesis and the finished game. This is the main thesis which provides information on the entire project. In addition to this document, the game had to be completed and delivered with it. In total around 241 hours were spent on the thesis and 116 hours were spent developing the game.

Throughout this project, some deadlines were missed. An important deadline missed was the first deadline for the pre-project plan and the vision document. The deadline for these documents was January 27th, but the documents were delivered on March 27th.

Another important deadline was for the initial prototype of the game, March 30th. The prototype was not complete for this deadline and was not met. The first prototype was finished for the meeting on April 13th.

Throughout the development process, the agile development methodology FDD was followed. During the start of the development process, an understanding of the requirements, scope, and general information of the game was acquired. Normally this would be used to create a domain model of the system, but this was not relevant or useful in this context, and instead, the understanding of the system was better described through feature requirements and user stories. Building a feature list is the second part of the FDD model and was followed in this project. Feature requirements were written down as user stories in the documentation. The next parts of FDD are iterative and were done iteratively through multiple sprints. Planning, designing, and building features were part of every sprint. These phases were applied for every implemented feature, with more focus on building as it was the most time-consuming. The planning phase was used for understanding how to implement a feature well and for good code implementation. The design phase was used especially for the user interface and for all the models. The build phase was the most ubiquitous and time-consuming and was found in every aspect of building all features. The sprints were two weeks long and allowed for several features and significant developments to occur between them. After every sprint, meetings were held with the advisor to verify progress and discuss further features to be implemented. These were added to the requirements documentation with more user stories.

This project had 3 distinct sprints. The FDD methodology was used, so the focus for every sprint was to add new features. In the first sprint, the main added features were the maze and pellets. The player could walk through the maze using the Omnideck and walk into pellets in the game to collect them. In the second sprint, the main added features were the ghosts and their behavior. These would spawn in the maze and chase the player until they collected a power pellet, at which point they would get scared and run away. In the third and final sprint, the main added features were the items. These could be walked into then grabbed by the player and used against the ghosts in various ways.

Some additional administrative goals were outlined in the pre-project plan describing how and when work would be done, along with personal improvement goals. One goal was to work regularly throughout the entire project instead of at the last minute and divide working hours effectively. Work was done throughout the entire project, but the last month of the project saw the most work hours completed. This goal was partially achieved. Another goal was to improve at logging necessary processes during the project. Throughout the project, several processes needed to be documented, and this was achieved.

4.2 Development Results

The overarching development goal for this project was the development of a demo game for the Omnideck that closely resembled Pac-Man in a VR setting. This goal can be split up into several smaller regarding functionality, usability, reliability, performance, support (FURPS), development, and overall enjoyability of the game as was described in the pre-project, vision, and requirements documentation. (Pre-Project Plan)

All required functionalities are listed in the requirements documentation (Requirements Documentation) in the form of user stories. Functionality requirements can be split up into technical and design requirements. Technical functionality requirements are about what the game and system are technically capable of while design functionality requirements are about the mechanics and possibilities inside the game. The most important technical functionality requirement was being able to play the game in VR while using the Omnideck as a motion controller. There were several important design functionality requirements instilled to make the game stay true to the original Pac-Man. Some of these were Pellets, Power-Pellets, obtainable fruit, a score, teleportation, 4 different ghosts, 4 different ghost states, and accurate ghost behavior. In addition, other design functionality requirements were in place to fully utilize the VE and add creative features to make the game more fun. Some of these were different items obtainable from fruit, a toggleable map, and a scoreboard. All these requirements were fulfilled in the final version of the game.

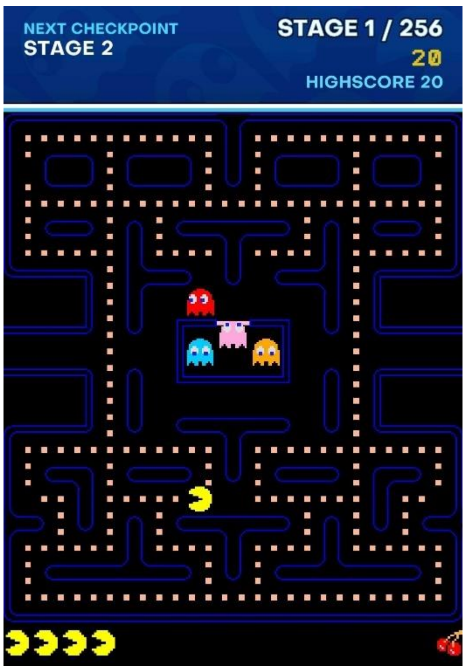


Figure 14: Screenshot of Pac-Man mobile game for reference

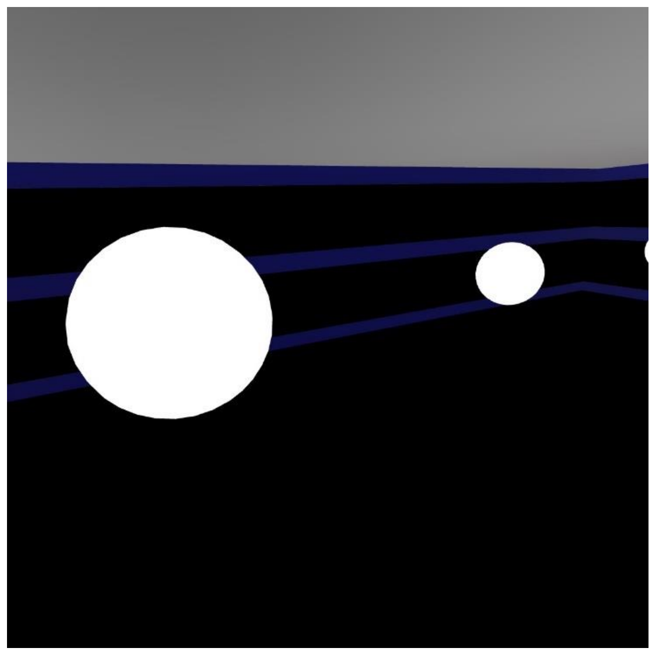


Figure 13: Power Pellet (left) and regular Pellet (right) in the VR game

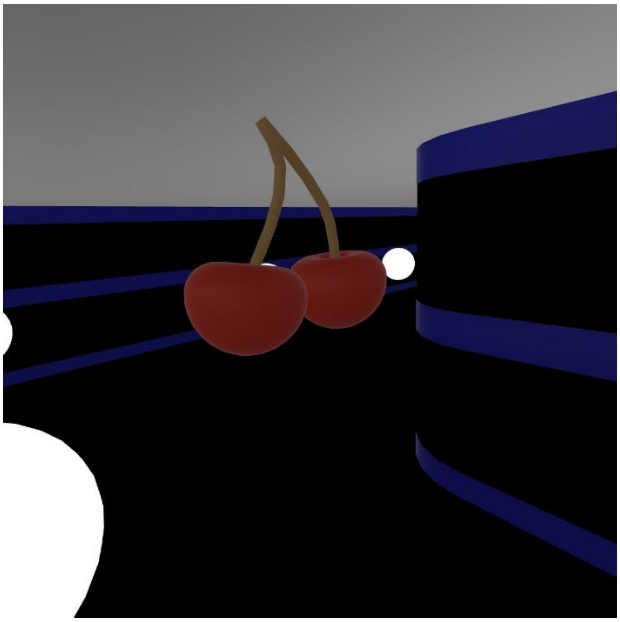


Figure 16: One of the obtainable fruits in the VR game

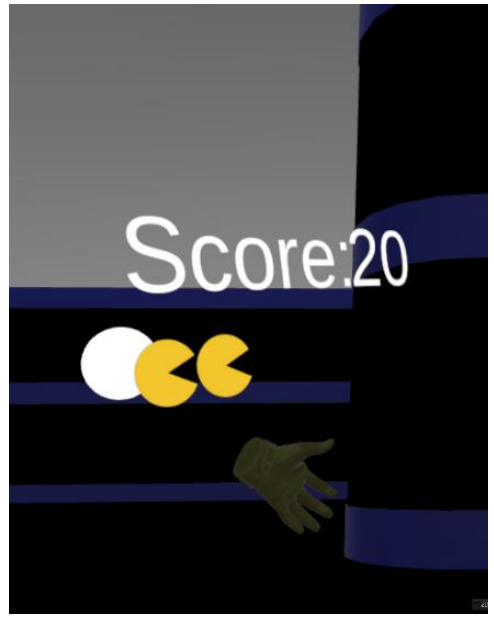


Figure 15: Score displayed in the VR game

Figure 18: Teleportation wall in the game, walking through
teleports the player to the other side of the maze

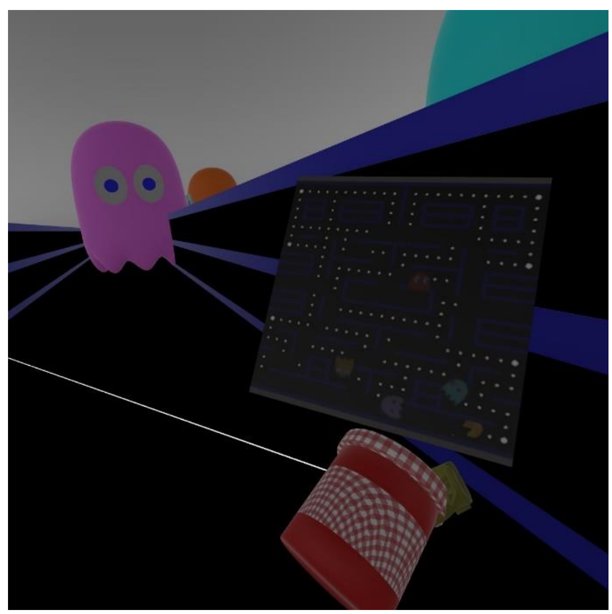


Figure 17: Multiple ghosts shown in the game along with a held item and the map

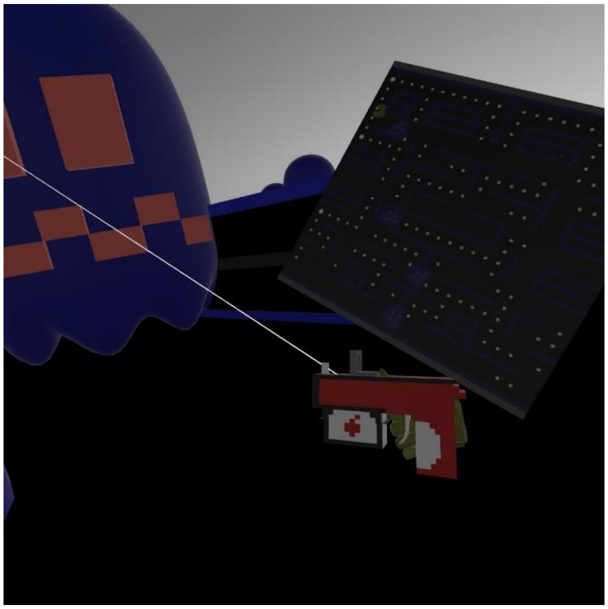


Figure 20: A cherry bomb gun inspired by Pixel Gun 3D, drops from the cherry item

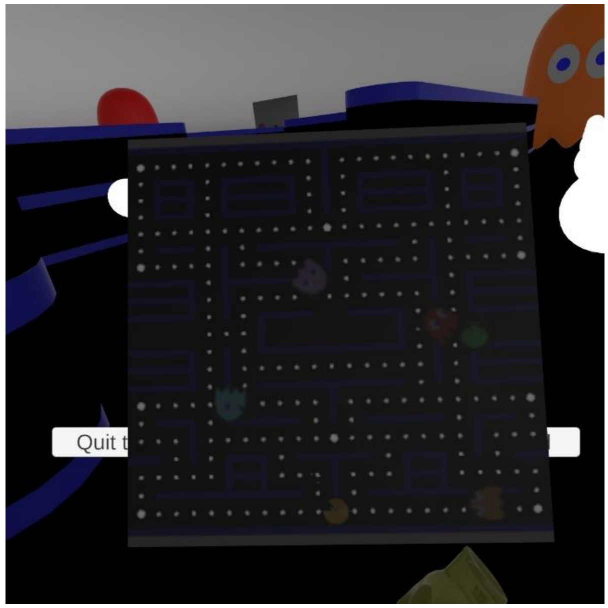


Figure 19: Map of the maze shows the player, ghosts, pellets, and items updated in real-time

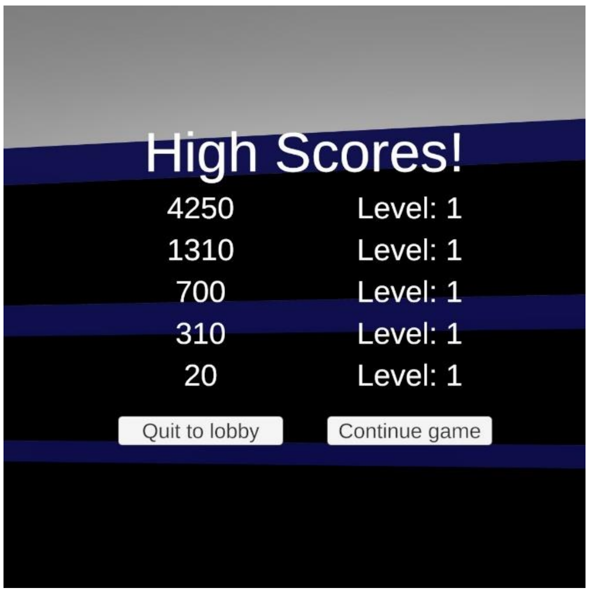


Figure 21: High score table displays after dying

Usability requirements included having a simple, understandable, and intuitive user interface and controls, and the game should be consistent and responsive. The user interface was designed with simplicity and intuitiveness in mind, with few, but essential buttons, and simple navigations between the two user interface pages. These pages follow the users head tracked movement, so they do not get lost in the VE, and the buttons are large and descriptive. Controllers are simple to use and use a familiar layout. There are only three buttons used on the controllers and actions were mapped to the most intuitive buttons, however, there is no in-game information on how to use these controllers. Motion was controlled through tracking devices and the Omnideck which is simple to use as only walking is necessary. The overall usability requirements were mostly met in the final version of the game.

Reliability requirements were generally about how reliable the playing experience is, for example, if the game crashes or has severe glitches, it is unreliable. No crashes happened throughout the entire development process or in the finished product. However, several glitches and bugs were found throughout the development of the game. These were fixed as they were found. The most common reliability issue occurs with tracking errors or disconnects between the different software which is necessary for running the game with the Omnideck. These errors can be severe and affect user experience and gameplay, but these are isolated and separate issues unrelated to the game and not considered for reliability requirements. Reliability requirements are met in the final version of the game, but as with any game, more bugs and glitches could be found in the future.

Performance requirements regard the technical performance of the system while the game is playing. One important performance metric, especially for VR, is FPS. Throughout the game development process, performance was taken into account as it is vital for a good VR experience. This included making lower poly models in Blender and using less expensive calls in scripts to avoid performance issues. The game runs with no noticeable performance issues so performance requirements are fulfilled in the final version.

Support requirements included the maintainability of the game. This was taken into account in several ways from the beginning of the project. The chosen version of Unity was 2021.3.16f1 which is a long-term support version of Unity. The project used Git extensively to save progress and to upload the game to GitLab, where a large file storage system was used for large files to improve maintainability. While writing code, variables, methods, and classes were named based on standard naming conventions with names corresponding to their function. The code was documented to ease potential future development, and there is a readme file for the project which thoroughly explains the system on GitLab. The support requirements were fulfilled in the final version of the game.

Development requirements regard the development techniques used and following standard programming best practices. Throughout the development process, an agile-adjacent work methodology was used to improve productivity and speed of development. Many programming best practices were followed such as documenting, naming conventions, keeping the code simple, reusing code where possible, portability, and scalability. However, a specific and controversial debate exists in the unity development community regarding the best practices of singletons and managers. Some entirely avoid singletons and managers as they can cause poor scaling, difficult to read, and hiddenly linked code which can be very difficult to untangle in large-scale projects. Others use singletons and managers as they greatly reduce the development time and on a small scale are simple to use. In this project, some singletons and managers were used, but it is unclear whether this follows the best practice or not, as the best practice in this regard is controversial and undecided. Development requirements for this project were mostly fulfilled in the final version of the game.

Lastly, and possibly most important, was the enjoyment factor of the game. This is difficult to quantify or evaluate as the enjoyment someone gets from playing the game is entirely subjective. However, many considerations were made to make the game more fun, and many users were eager to play it of their own volition. I also personally enjoyed playing the game.

4.3 Omnideck Findings and Research Results

While studying the Omnideck multiple unnatural interactions were discovered. These mostly regarded human interaction with the Omnideck on a physical level, but some were also software related. This section is split by the found interactions, and each is further split by the three research questions.

4.3.1 Walking Straight

While walking in a straight line the user might experience a loss of balance or a swaying motion while walking. This swaying or stumbling could feel like walking while slightly inebriated. The origin of this problem is not entirely certain, however, there are multiple possible explanations. One cause could come from the user stepping on multiple rollers with different angles at once.

Figure 22: Foot between rollers

When the user steps on rollers with different angles, the foot will experience forces in two slightly different directions, which is unnatural. These incongruent forces are enough to cause a slight but noticeable imbalance. They might not be strong enough to trip the user but can change their footing placement and direction, which can make the user walk into the gap between rollers again. This could also lead to the second possible cause, where either foot is on a separate segment of rollers.

Figure 23: Feet on separate segments

When the user walks on separate segments of rollers with each foot, this causes each foot to have a different direction and be pushed backward in separate directions, which is unnatural. This can cause a slight sway, change the direction of walking, or lead to one of the other described scenarios

Both scenarios come innately from the Omnideck's mechanical design. Every segment of the deck has rollers angled towards the center and will exert a force on the user's foot in different directions from other segments.

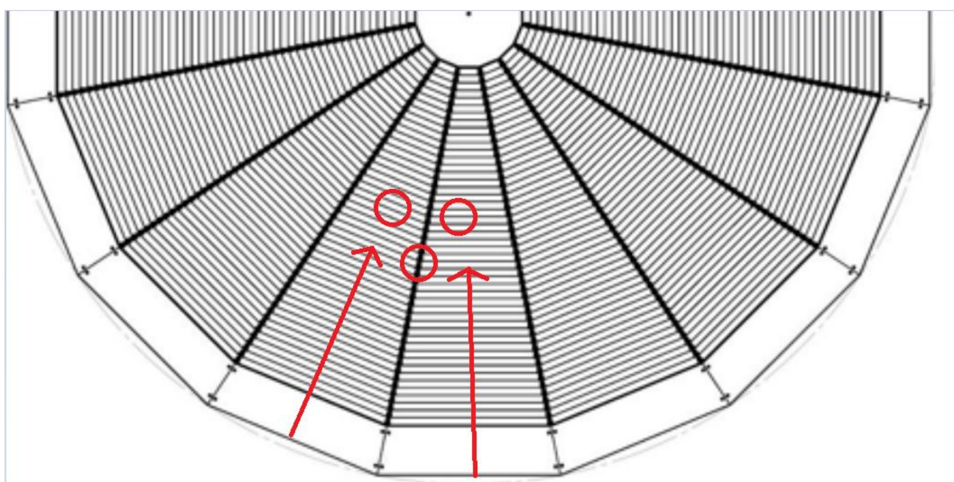


Figure 24: Top-down view of force vectors (red arrows) and foot placements (red circles)

Lastly, even while walking on one segment exclusively, the user might still experience some imbalance. This could be due to the Omnideck pushing the foot towards the middle regardless of foot placement which contrasts with a natural walking pattern where the foot pushes directly backward. This becomes more obvious during turning.

Figure 25: Foot on an exaggerated angle,
The F1 vector shows the foot's natural backward movement,
The F2 vector shows the Omnideck segment's force vector

This imbalance varies in severity and could range from a slight and barely noticeable imbalance to the user stumbling forward or potentially tripping in a worst-case scenario. This also causes the user to experience a less than natural walking movement due to the irregular forces applied to the feet and could potentially lead to lower immersion.

This happens inherently due to the Omnideck's mechanical design, and thus workarounds might be more difficult to implement. Displaying the different segments of the Omnideck visually for the user would allow the user to avoid stepping on multiple segments at once and thereby reducing the severity and risk. This could help because, with the current system, the user has no way to sense how many segments or which part of the segment they walk on other than sensing the forces through their feet.

4.3.2 Turning

Another unnatural interaction arises when the user turns and walks in a different direction on the Omnideck. When the user walks on the rolling segments of the Omnideck and initiates a turn, they will begin walking "against the grain" from the rollers.

Figure 26: Feet sideways on the Omnideck while turning

While walking in this manner the user will experience a force pushing sideways on their feet. This force is unnatural compared to natural walking and can cause a loss of balance or stability for the user. This has the potential to cause larger problems such as tripping or falling, especially at higher roller speeds and further toward the edges of the Omnideck.

Multiple solutions for this limitation were tested. To decrease severity, the user walked at a lower speed closer to the center during the turn and returned to a normal walking speed. When the user is closer to the center and walks slower the roller speed decreases and lowers the severity of the limitation. Another attempted solution for sharp turns was for the user to avoid turning on the rollers altogether, and rather stop before a turn, wait to be rolled to the middle, then proceed walking in the desired direction. Though this does eliminate the difficulties with turning, it is arguable that the user is no longer walking or turning naturally. A last method was tested where the user walked through longer, less sharp turns which decreased the severity of the limitation. By having less sharp and longer turns, the users walking more resembled walking straight and eased the difficulties of turning.

4.3.3 The Center

Another inherent feature of the Omnideck is its immovable center. The important distinction between the center and the rolling segments is that the center does not move physically and while the user stands in the center. (Chapter 2.5.2). This means that while in the center, the user experiences fully natural locomotion, compared to semi-natural locomotion while on the rolling segments. If the in-game movement speed is poorly adjusted, the difference between walking on the rollers and in the center become even more apparent.

Figure 27: One foot on the rollers, the other in the center

This is most noticeable when the user turns 180 degrees and walks through the entire center. This means the locomotion state changes from semi-natural to fully natural then back to semi-natural within a short frame of time. A potential solution could be to avoid 180-degree turns. If a 180-degree turn is necessary, then allowing the user to stop in the center before continuing will delay the switch between the locomotion types.

4.3.4 Speed Regulation

Walking speed and speed inside the game turned out to be important characteristics that could cause limitations for the Omnideck. The Omnideck rollers have two speed modes, and the roller speed adjusts automatically based on how far the user is from the center. The in-game speed is separate from the roller speed and can be adjusted in the Omnitrack application by increasing or decreasing the translation factor.

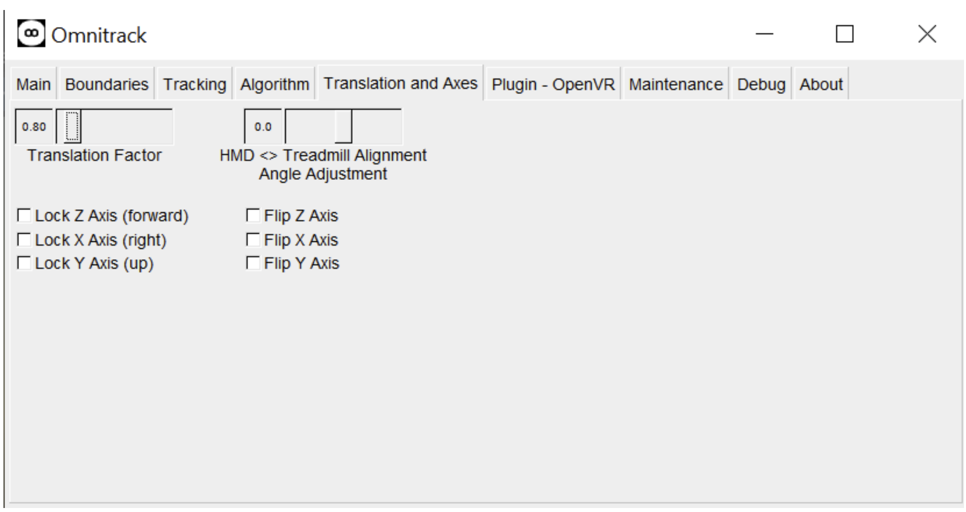


Figure 28: Omnitrack software, translation and axes settings tab

However, this translation factor is within the proprietary software and not accessible in the game code. This means that to change the in-game speed, a user must walk off the Omnideck and slide the translation factor scale. This was tiring for users, who had to take off the HMD and then walk on and off the Omnideck multiple times to test if the speed was too fast or too slow.

If the user feels the in-game speed is too slow, they will naturally walk faster to increase their speed. This can cause the user to walk faster than the rollers spin and can lead to them walking off the rollers and onto the non-moving skirt part of the Omnideck. This sudden change from high-speed moving rollers to a non-moving part trips the user.

Figure 29: Foot hits the Omnideck's outer edge

This was among the most severe limitations found. Walking off the edge onto the skirt had the highest impact on the user's stability and balance and caused tripping every time it was tested. The inability to change the speed multiplier while staying on the Omnideck was also a limitation as it created too many extra steps for the user.

Luckily, there were multiple potential solutions found for this problem. First, by adding a speed multiplier in the API, the speed could be adjusted to the user's preference in the game. Doing this made it unnecessary for the user to walk as fast on the Omnideck, thus reducing the risk of overstepping. This also can increase presence as the speed will feel more correct to the user when adjusted properly. A second solution is to add a visible marker around the outer edge of the Omnideck like the one around the center. A visible limit will make the user aware of the outer skirt, so they do not step on it. A last potential solution is adding a speedometer to show the maximum allowed walking speed. Since the speed is based on the distance from the center, a maximum speed will be reached at the outer edge, if a visible speed limit is shown this could prevent the user from exceeding the speed necessary to step onto the outer skirt.

4.3.5 Head Turning

An essential part of VR is the environment surrounding the user, so the user will move their head to look around, however, this is somewhat limited while walking on the Omnideck. While walking forward on the Omnideck, it can be more difficult to look fully around in the VE and the range of motion for the user's head could be more limited. Looking around at more drastic angles could also lead to a loss of balance, especially in combination with any of the other limitations mentioned. The danger of losing balance or tripping was not as high as with other limitations, but in combination with other potential limiting factors, the risk can increase.

Figure 30: Looking around while walking forward

Lower walking speeds can decrease the severity of this limitation and will allow the user to more freely look around. Another option is to let the user take a break from walking to look around while standing in the center. This completely removes the need to look around while walking.

4.3.6 Tracking/Crawling

Tracking problems did not occur due to the Omnideck but occurred in unusual positions like crawling. If the tracking bases are set up incorrectly, or the wireless tracker on the Vive HMD is positioned poorly, tracking capabilities can be negatively affected. This can result in a wide range of undesired behavior from the Omnideck. Some of the recorded problems with a loss of tracking included failure to start, failure to stop, recentering, rolling the wrong segments, abrupt changes in rolling speed, and sudden starting and stopping of the rollers. A loss and worsening of tracking was also noticed while crawling on the Omnideck.

Figure 31: Crawling on the Omnideck

These erratic behaviors are dangerous and can cause a user to fall both due to sudden changes in roller speed and movement, and due to the mismatched, poor, changing, or erratic visuals given through the HMD. The severity of bad tracking can be high and unpredictable. To prevent a loss of tracking, correct setup of the tracking devices as described in the user manuals is necessary.

4.3.7 Terrain

The API allows the in-game character to walk up and down inclines, however, due to the design of the Omnideck, it cannot create physical slopes. The Omnideck has no way for the user to walk on a physical slope so there will be a disconnect between the game world and real world. This disconnect is noticeable by the user and could lead to increased cybersickness and decreased immersion due to the sensory conflict the user experiences. To avoid this, slopes were avoided as they were not required for Pac-Man.

4.3.8 Noise

While walking on the Omnideck, the rollers spin which causes sound to be emitted. This sound is claimed to be quiet enough to be used in an office environment. (Omnifinity, n.d.).

The noise level increases at higher speeds and stops when the user stops walking. The noise level ranged between around 50dB to around 75dB. The lowest sound level could be compared to a quiet library while the loudest could be compared to busy traffic. This can be distracting or decrease inclusion.

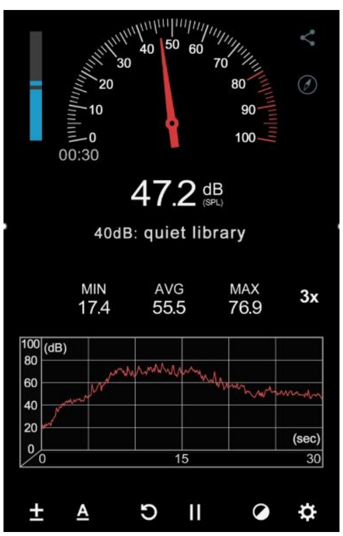


Figure 33: Sound level meter indicating 47.2 dB

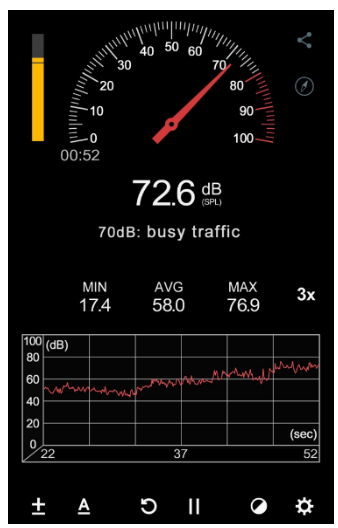


Figure 32: Sound level meter indicating 72.6 dB

A way to counteract this is to play sounds and music in the game which drowns out the noise made from the Omnideck. Another way is to walk slower on the Omnideck as slower speeds produce less sound. As a last resort, earplugs could be used to block the noise.

Chapter 5: Discussion

Two large constraints heavily influenced this entire project, the group size and time constraints. Although most bachelor's projects began with the semester start in week 2 of 2023, I was on an exchange semester abroad that lasted throughout February. This meant that the project only started at the beginning of March, despite some initial meetings that happened earlier, and this meant that the project had to be done with no other group members. This massively decreased the amount of time available for this project, which also decreased the possible workload. This was compounded by there being only one group member, which further reduced the work capacity for this project. This was a major limitation throughout the entire project and reverberated for every part of the project. It strongly influenced decisions made in favor of speed, time efficiency, and reduction of workload for all tasks.

5.1 Administrative

The administrative tasks and processes were important to follow during this process and despite the severe constraints, were upkept.

Two deadlines were missed in this project as explained in chapter 4.1. The first deadline for the pre-project plan and vision document was missed due to studies and exams abroad. The deadlines were set before this project even began and were rescheduled for March 17th and were met. The second unmet deadline was for the initial prototype, as this was an ambitious and intentionally early deadline. The original deadline was in the middle of easter when no meeting could be held. Therefore, an ambitious and early deadline was put in place with the expectation of potentially not meeting it and falling back on a second deadline after easter. When the deadline was not met, a new deadline for the week after easter was installed and thereafter, met. Missing the first deadline was unavoidable, while missing the second deadline was prepared for in advance, so missing it was insignificant.

Some documents through mutual agreement with the advisor were deemed unnecessary for this project. These were the work contract and GANNT diagram. Since this project only had one participant. And there were no external stakeholders outside of NTNU, so a work contract was unnecessary. The GANNT diagram was discarded in favor of a milestone plan. This was chosen due to its simplicity, speed, and since it lowered the administrative workload while showing the same deadlines.

Following FDD methodology worked well with the given constraints. The FDD methodology was lightweight with few requirements, which lessened the administrative workload while providing a good framework for the quick development of new features which was desirable. There were some difficulties using this method, specifically with following the method exactly and using multiple agile techniques. The first step of developing an overall domain model is better suited for developing other applications than games. Games span several domains constantly and can be better modeled through, for example, flowcharts or storyboards. Instead of creating models for this step, an overall understanding of the requirements was obtained which was then used in step two to make a list of required features. The rest of the methodology was followed closely with a larger focus on the development and testing of features rather than meticulous planning since this sped up the development process. Some agile techniques were used, but many provided no additional value or were infeasible due to the group size constraint.

Some administrative goals in the pre-project plan could have been performed better, but the goals were achieved. The first goal was to work evenly throughout the entire project. Work was done throughout the entire project and working hours were spread evenly in the context of the entire working period given that it started in March. However, the spread of working hours in between weeks varied significantly and there was a clear increase in working hours towards the end of the project. The irregularity of working hours per week mostly involves lab availability during the development process and personal life. Here, the group size constraint also comes into play, as the working hour irregularity will be higher with only one group member compared to multiple. It is also clear that more working hours were needed towards the end of the project for the main thesis and documentation. This is an uneven spread of working hours and shows that the thesis could have been started earlier to more evenly spread out working hours. However, the time spent was generally well spread out among different tasks and throughout the entire process.

5.2 Development

The final product fulfilled all requirements from the requirements document, pre-project plan, and vision document, it lived up to the stakeholder's wishes and expectations, and serves its purpose as a tool for research and as a demo game for demonstrating the Omnideck's capabilities at the Visualization Lab.

Several important design decisions were included to make the game more closely resemble the original Pac-Man (Chapter 4.2), and some design decisions were made to help solve some of the found limitations. One of the most important implemented design choices was the speed controller. This was initially implemented because players were moving too slowly in the game, but it inadvertently ended up being a potent solution for the fourth limitation and was one of the most valuable changes made in the game. Creating a speed multiplier allowed the user to select their preferred speed, but also allowed for more game mechanics to be introduced that were previously impossible since the player was limited to a single speed. Including this speed controller was an essential design choice and the game would be substantially worse without it.

Another design decision was to increase the ghost's height after increasing the height of the walls. Initially, the walls were lower than the player so they could see where the maze went, however, the wall height was increased, and a map was added for navigation. This caused the fifth limitation to become prominent, so ghost height was also increased to incentivize looking around more.

Some of the design solutions found from research question three were not fully implemented but could have led to better game quality. One example of this is the solution for the first two limitations. A fully segmented display of the Omnideck was never implemented mostly due to the time constraint, instead, the basic blue circle from the Omnifinity prefab was kept that showed the center of the Omnideck. Creating longer turns was another found design solution that was not implemented since it was more important to keep the maze close to the original Pac-Man which only had right-angled corners. Changing the map would make it stray too far from the original and the map was too big to be upscaled further.

The technologies used for creating the game worked seamlessly together and contributed to development speed. The Omnitrack API was simple to implement and worked seamlessly with the other technologies. Getting the Omnideck to work with Unity, SteamVR, and the HTC Vive was effortless, all that was required was to use the prefab from Omnifinity's GitHub page and turn on all necessary applications.

Using Unity also sped up the development process since it has many active users that make tutorials and code snippets that were helpful in creating the game. This helped multiple times when problems were encountered during development that would have taken time to figure out alone.

Using the FDD methodology during development helped increase the number of features in the game and worked well despite the constraints. Many agile methodologies would have been more work to do with little benefit since they were made with larger groups in mind, but the FDD methodology focusing on producing features worked well within the context and constraints of this project.

5.3 Research

The research in this project had multiple limitations that came from the time and group size constraints. One of the most important limitations of this research was the scope and amount of research conducted. There are multiple things not accounted for in this research that would be valuable to know about using the Omnideck. It would be valuable to know how this semi-natural locomotion technology compares to both natural and super-natural techniques, more in-depth on how it affects immersion and presence, and if it increases or decreases effectiveness. The time constraints severely limited the scope of the research done, which in turn restricts what this thesis can answer. These constraints also limited the types of research methods used. No large-scale user tests were done, which is a clear limitation. For more unbiased and structured data, quantitative data from several user tests and experiments would be necessary, which would give a much clearer picture of the severity of limitations. In addition, more research strategies and data generation methods should have been used. This affected the second research question and as a byproduct the third research question. With more research and data, the second research question could be answered more thoroughly and could lead to better solutions for the third research question. Despite these limitations, the research is still valuable as true limitations can be shown by showing flaws in the natural interaction between the Omnideck and the user. These flaws were proven to affect the user to some degree and solutions were implemented that improved performance.

This project started with broader, expanded, and more nebulous research goals especially for the second research goal. This included to find the severities of the found limitations by doing experiments and several structured user tests and comparing cybersickness and discomfort levels. However, this was cut out due to the constraints.

5.3.1 RQ1

"What limitations exist with the Omnideck?"

The results from user tests, evaluation, and data analysis showed conclusively that there are flaws in the interaction between the users and the Omnideck, and that these are limitations that arise mostly from the Omnideck's design.

Of the eight limitations found, six could be traced back to how the Omnideck is designed. The first three limitations occur due to the way the rollers push the user's feet. The rollers always exert a force pushing towards the Omnideck's center on the user's feet, regardless of their direction of travel, rotation, or angle. This is an unnatural force that never occurs for completely natural walking. This force is the root cause for the first three limitations of the Omnideck. The fourth limitation also has to do with the Omnideck's design as nothing is stopping the user from walking onto the skirt or off the edge of the Omnideck. The Omnideck does stop rolling if the HMD goes too far from the center, but often this happens after the user has already walked onto the skirt or off the edge. The seventh limitation is the inability to physically create a slope for the user. This one was not noticeable in the game as all the walkable areas are flat, however, this could be a problem in a game or application where walking on slopes is a requirement. The eighth limitation is the sound level. Whenever the user walks on the Omnideck, it will make noise between ~50dB to ~75dB which is loud enough to be noticeable and potentially be a disturbance to the user. This noise level can be compared to the weight of the headset as it is an ever-present reminder of the real world that could decrease inclusion.

Two more limitations are not directly caused by the Omnideck's design and occur due to second-hand factors.

The fifth limitation was about head turning and was more of a result of the culmination of the previous limitations. This is already an abnormal task, but with the added unnaturalness of the Omnideck, it is more difficult and avoided by users.

The sixth limitation is unique as it is not a limitation with the Omnideck itself. This limitation came from how the Omnideck behaves after losing tracking. When tracking is lost, the Omnideck can start to behave erratically. Different segments can start spinning fast for varying lengths of time and it becomes a risk for users to stand on it as they could fall. It is unclear exactly in which link tracking is lost, but the Omnideck's behavior when it happens can be called a limitation.

5.3.2 RQ2

"How can these limitations affect the user?"

This research question goes hand in hand with the first as the limitations found with the Omnideck often are limitations because of how they affect the user. The results from the user tests show that the limitations can affect the core parts of the user's immersion and thereby affect user presence. Beyond that, it was shown that unnatural walking could pose a risk of losing balance, tripping, and falling. When deeming whether the limitations affect the user's experience, the core parts of VR immersion are looked at. If a limitation affects inclusiveness, extensiveness, vividness, or the surrounding or matching essentials of immersion, then it can affect the user negatively and decrease presence.

The first three limitations can negatively affect vividness. Specifically, interaction fidelity could be lowered due to the semi-natural nature of the Omnideck. When a user walks, they experience unnatural walking while walking forward and especially while turning. While walking forward they might experience a swaying motion from side to side or struggling to walk in a straight line as if they were slightly inebriated. This has to do with the semi-natural interaction fidelity that the Omnideck provides from the unnatural force from the rollers exerting a force on the user's feet. This can negatively affect user experience and decrease presence from the reduced immersion caused by less than natural fidelity. The quick change between natural and semi-natural locomotion while walking over the center also caused the user to feel a momentary decreased presence from the inconsistent interaction fidelity.

The fourth limitation can negatively affect inclusiveness, vividness, and the match between user actions in-game and real life. While walking, the user might feel that their walking speed is faster or slower than the in-game movement speed. This makes the user feel like there is a mismatch between their actions and what happens in the game and can negatively affect their presence. This also falls under the category of vividness and interaction fidelity because the user feels that the walking speed is not lifelike, or the same as their walking speed. If the in-game speed feels too slow, they will increase their walking speed and walk further away from the center. If this causes the user to walk onto the non-moving skirt, their foot suddenly stops as they are no longer walking forward on the rollers which breaks inclusiveness and immersion. In addition, this change from walking to stopping is abrupt and unpredictable for the user and can cause them to lose balance, trip, or fall.

The fifth limitation can negatively impact the surrounding nature of the VE. All the previous limitations combined culminate in an environment where the user needs to focus more on walking both to walk correctly and avoid losing balance. In this environment, it is more difficult to look around in the environment to experience the surrounding nature. One example of this is that the user will avoid turning their head around to avoid losing balance. This takes away from the surrounding nature of the VE since it lowers its surrounding nature and can cause decreased presence for the user.

The sixth limitation can affect vividness and the match between the user's actions and sensory stimuli. If tracking is lost or the quality is decreased, it can cause erratic signals to be sent to the Omnideck or the game and can cause both to start moving randomly or erratically. The Omnideck’s erratic behavior is potentially dangerous because it includes the rollers spinning fast and randomly. If the user is standing on the rollers while this happens it has a severe risk of falling on the metal rollers with no warning or indication that it will happen. During certain user tests a loss of tracking happened where the test had to be stopped immediately as the user no longer felt safe on the Omnideck. This has a strong negative impact on vividness due to the lack of consistency. In less severe cases the Omnideck can receive incorrect tracking data as if the user is positioned in another location. This can cause the Omnideck to send incorrect movement vectors and the in-game character will move the wrong way compared to how the user moves. This also strongly negatively affects the match between the user’s actions and sensory stimuli as the character will not move based on the user's movements. Lastly, a loss of tracking was noticed while crawling on the Omnideck due to the irregular user position, in this case, tracking was completely lost so no movement occurred in the game and the rollers did not move. Despite this limitation not being caused by the Omnideck, common tracking and sensor errors such as drift delay and inconsistency all produce adverse and undesired effects for the user which are strengthened by the use of the Omnideck.

The seventh limitation could affect extensiveness and the match of user actions and game actions. The Omnideck cannot increase or decrease the slope angle like conventional treadmills. However, this did not affect the user in this game, as Pac-Man does not have any slopes, so they were not required. However, in certain games or applications, it could be necessary for the user to walk on slopes. When the user walks up a slope in the game, the increase in angle will not be reciprocated by the Omnideck which causes a user sensory conflict. This also is a limit to the extensiveness of the Omnideck as it is unable to reproduce slopes. However, this did not affect the user in any way in this game as Pac-Man does not have any slopes, so they were not required.

Lastly, the eighth limitation comes from the noise level of the Omnideck. Like the weight of the HMD, the noise is an ever-present reminder of the real world, which is undesirable when it comes to inclusion. Optimally the Omnideck would not make any noise and the HMD would not weigh anything, but this is impossible. Regardless this can affect the user's sense of inclusion and thereby presence negatively.

5.3.3 RQ3

"Can these limitations be counteracted?"

Despite many of the limitations originating from the innate design of the Omnideck, several limitations can be counteracted by making conscious design choices that either enhance perception or change the behavior of the user.

The first limitation can be reduced by displaying the different segments of the Omnideck virtually for the user. Displaying the different segments allows the user to better align their walking with a specific segment of the Omnideck, reducing the level of unnaturalness the more aligned they walk with the segment. Something as simple as colored lines to represent the different segments of the Omnideck would be adequate. This can have the disadvantage of decreasing inclusiveness since the user is reminded by the outline that they are walking on the Omnideck.

Despite the second limitation being caused by the same mechanics as the first, it cannot be mitigated the same way since the goal is for the user to turn. There are multiple ways to counteract the limitation. One way to counteract this is by simply avoiding turning on the rollers. If the game is designed in a way where the user stops before each turn, the user will only turn after they have been rolled back to the center of the Omnideck. This eliminates the problem that is caused by turning on the rollers since the user no longer turns on them but will often be an undesirable solution as it can impair the pacing of the game. Another solution could be to design the game in a way where the user walks slowly on the rollers while turning. Reduced walking speed reduces roller speed and reduces unnaturalness. This is similar to the previous solution with a slightly different tradeoff, instead of eliminating the limitation, the effects of it are reduced. Having to take slow turns can also impact the pacing of the game if walking otherwise is fast-paced. A third solution could be to create longer turns. The limitation of turning on the rollers comes from the discrepancy between the angle and direction of the foot compared to the force exerted by the roller so minimizing this would increase naturalness. Limiting the turning angle would decrease this discrepancy and minimize the limitation. This would also allow for higher walking speed during turns, which is often desirable. This limitation was put on display in this game, since Pac-Man has several short right angles, and the user has to constantly turn to navigate the maze. When such short right angles are necessary a last resort could be to create a larger virtual game area which in turn would create longer bends and more leeway for the user.

The third limitation comes from the frequency of changing between semi-natural and fully natural locomotion, therefore the solution comes from decreasing the transition frequency. Allowing the user to stop in the center before turning around and continuing was the main solution found to this limitation. This can be achieved through game design where the user comes to a complete stop before turning around 180 degrees. In Pac-Man this would have been difficult to implement and would reduce freedom of movement for the user in a real-time game environment, however, this would be a reasonable solution in a story or puzzle type of game with different rooms or segments.

The fourth limitation can be solved by allowing the user to adjust the speed in-game and showing where the boundaries of the Omnideck are. The problem in this case stems from the user's desire to go faster in the game combined with the lack of information about the boundary of the Omnideck. By outlining the boundaries of the Omnideck in VR, like the solution for the first limitation, the user knows not to cross it and the problem is avoided. This does have the disadvantage of decreasing inclusiveness as the user is constantly reminded of the boundary and Omnideck that they are walking on, which are outside of the VE. Another solution is to give the user control over their speed in the game. The user wants to go faster so they will naturally walk faster to increase speed which leads to walking over the edge. Instead, by increasing in-game movement speed, they do not have to walk as fast on the Omnideck to move their desired speed in-game. This solution has the advantage of better matching the user’s real-life walking speed while simultaneously not affecting inclusion. Another solution is to create a speedometer for the user to show the maximum speed they can walk without crossing the edge of the Omnideck. This is possible since the user's walking speed is directly increased by distance from the center, so at the edge the user will walk at a given speed which acts as a speed limit. This has the benefit of giving the user a clear boundary without the decreased inclusion that can affect presence. Multiple of these solutions can be combined to further reduce the chance of hitting the edge of the Omnideck.

The fifth limitation is caused by the combination of other limitations and the need for increased focus on unnatural walking, making it more difficult for the user to experience the environment's fully surrounding nature. The core of the problem is the user’s hesitation to look around while walking, which can be solved either by removing the need to look around to experience the environments surrounding nature or by enticing the user to look around through design decisions. One possibility is to allow the user to stop or lower their walking speed to have breaks, so they have more time to look around. This occurs in the game after the user collects a power pellet and the ghosts run away from the player. Designing solutions for this limitation can be unique to the game. In this game, the ghosts were made taller than the walls to incentivize the player to look around for the ghosts even when the player runs away from them.

The sixth limitation is not caused by the Omnideck or user interaction, so creating design solutions is challenging. The best way to avoid tracking problems is to prevent them from occurring to begin with. Correctly setting up all tracking components according to the user manual will reduce the chance of tracking errors occurring. Beyond this, a solution would be to implement code to detect when a tracking error occurs and stop the Omnideck from rolling when this happens.

The seventh limitation is innate to the Omnideck and is challenging to solve through design decisions, but the limitation can be avoided by creating flat maps where the user only walks on flat areas. Specific design solutions would vary from game to game, an example could be using an elevator to change floors instead of stairs. In Pac-Man slopes and changes in elevation are unnecessary and were avoided so this limitation never caused problems for the user.

The eighth limitation is caused by the noise emitted from the Omnideck. Solutions are to either reduce the noise level or drown it out. Drowning out the noise made by the Omnideck by playing music or sounds in-game can help reduce the negative affect the noise has on inclusion while improving extensiveness. Walking slower on the Omnideck also reduces the noise level, as slower spinning rollers make less sound than faster spinning ones. Lastly, a solution could be to block out the sound from the Omnideck is to use noise-canceling earbuds or headphones, or to use earplugs. This blocks out the sound and increases inclusion with the disadvantage of making in-game sounds harder to hear.

5.4 Reflection

This project has been a true challenge for me to complete alone, but the work I did was thorough and valuable. Completing an entire bachelor's project alone came with advantages and difficulties and was especially emphasized by the time constraint.

From the beginning of this bachelor's project in March, time was of the essence. I had no time for a break after several weeks of exams and immediately started working on the project alone with the knowledge that working alone and with a two-month delay from the beginning would leave me with far too little time.

However, in the beginning, working alone on the project provided me with some benefits. There was only light paperwork and a presentation to do before the development process began, which was not too difficult to finish alone. During the development process was where working alone was the most beneficial as it allowed for full creative freedom, and no difficulties with merging code, which is a notorious problem with Unity and Git, and it allowed for complete freedom in working hours. However, this also came with downsides. The largest downside was the lower workload that could be completed by a single person compared to multiple. Another downside is the lack of second opinions on the project. Even though the advisor did help with some decisions throughout the project in meetings, having a second group member developing and working would provide far more support and quality assurance for decision-making. Working alone on a large project is less motivational compared to working together with someone. This was especially noticed towards the end of development while fixing bugs and doing less amusing work.

After development, while writing this main thesis, the lack of group members is felt the strongest for several reasons. With fewer members, less work can be done, but the thesis still has the same required parts regardless of group size. This means the workload is increased comparatively to each member in a larger group. A second opinion is especially valuable while writing the thesis, which is missing when writing alone. Lastly, writing the thesis is not as fun as making the game, so it is more difficult to stay motivated while writing.

Despite these limitations, I believe the work I did was valuable for further research of the Omnideck, the game I made was up to the standards set, and I followed important processes throughout the project. I delivered a solid project and overcame the difficulties that faced me.

Chapter 6: Conclusion

All the research questions were answered with the findings in the results and analysis in the discussion, and they can help us conclude the problem statement. The Omnideck does not escape limitations, as multiple were found often pertaining to its design. The mechanics behind the way the Omnideck works create unnatural user interactions while walking. These unnatural interactions affect the core of immersion and can negatively affect the user's immersion and thereby presence. However, there are clear ways to circumvent the limitations by consciously designing games or applications with the limitations in mind. Multiple solutions to create games that counteract or avoid the limitations altogether were shown. To summarize, each research question can help provide a definitive answer to the problem statement.

6.1 Further Work

Further work would be valuable within this domain of research due to the limitations of this study. The limitations meant that the data provided has a risk of being less objective or incomplete. To amend this, research should be done on larger groups of users to test if the results persist, or if some limitations are less significant. This would also uncover more potential limitations that were not found in this study. Further research could be done to discover exactly how strongly these limitations affect user presence as this study was not able to quantify the significance of the limitations. Further work could also be done to find more design choices or techniques to further mitigate the found limitations and quantify the effectiveness of the design solutions. A better understanding of how significant each limitation is combined with design solutions that quantifiably mitigate them would make the priority of distinctive design choices more apparent.

Societal Impact

A factor of ever-increasing importance while working on projects like this one is to be conscious of the societal impact that can come from the technology. The impact this technology can have is multifaceted and there are several factors within multiple domains to be considered like environmental, health-related, economic, and multiple social factors.

As mentioned in chapter 2.3, VR applications and games are expensive to play. To play a game in VR, first, a powerful PC is required, in addition to an HMD, controllers, and two tracking stations. This is the bare minimum to run VR games, but there are further accessories and technologies available for purchase to improve the experience. This does not take into account the Omnideck which is too large and expensive for the average consumer to purchase. All of these technologies cost an exorbitant amount of money and are inaccessible luxury goods for most people. These are luxury toys that only privileged individuals are able to buy. Personally, I also don’t have a VR headset and had to borrow from the university throughout the entire project.

This poses economic social ethical issues, especially regarding equality. The United Nations' 10^{th} sustainable development goal is to reduce inequality within and among countries (United Nations, 2016). By having such a high economic barrier to entry, VR gaming is unobtainable for most people and does not help reduce inequality. This is especially true outside of Norway, where less developed countries have more severe economic and social problems and division, and between countries of disparate economic status.

There are also growing environmental and ethical concerns about VR technology and the computer industry due to the increased need for and use of electricity. To play VR games, powerful computers are needed for higher performance, which uses more electricity than conventional computers. Growing electricity use is a trend in everything computer related as they use more power, and more people use computers. This is compounded by the effects of the recent energy crisis, which has reduced the accessibility, affordability, and cleanliness of energy sources. (IEA, 2023)

This poses environmental ethical issues. The United Nations’ 7^{th} sustainable development goal is to ensure access to affordable, reliable, sustainable, and modern energy for all (United Nations, 2016). It is especially important for VR developers to keep this sustainability goal in mind to reduce the negative effects their games contribute to this problem.

Another ethical concern for VR technologies is their close ties to military use. Many new technologies often pick up the interest of militaries for their potential use for military applications. This has been the case with VR as mentioned in chapter 2.4 where the link trainer was used to train military pilots. VR technologies are still developed for and with militaries and are still used by militaries today. The Omnideck was considered for and used for military applications in the form of training soldiers for real-life combat scenarios. (Army Technology, n.d.) (Khimeche & Wolf, 2017).

This poses societal ethical issues. The United Nations’ 16^{th} sustainable development goal is to promote peaceful and inclusive societies for sustainable development, provide access to justice for all and build effective, accountable, and inclusive institutions at all levels. (United

Nations, 2016). VR technologies that are used for military purposes do not contribute to peace.

A final ethical concern that is especially pertinent to this project is gaming addiction. This is a health-related ethical concern that has been a growing problem with the improvement of games and the increase in the number of people who play games over several years. Internet gaming addiction was added to the DSM-5 by the American Psychiatric Association (APA) as a condition that requires further study. (Petry, et al., 2014). Excessive gaming could affect players' physical and mental health.

However, there is also good to be found in these VR systems. Simulations can have positive environmental, health-related, and economic effects.

Simulations are innately designed with economic and health-related factors in mind. A simulation is meant to increase efficiency and safety while decreasing the cost of experiences compared to real-life scenarios and training. The link trainer, as mentioned in chapter 2.3 is just one early example of this, expensive and dangerous pilot training was cheaper, more environmentally friendly, and completely safe compared to flying a real plane, while still providing valuable training. This is far more sustainable both economically, environmentally, and health-wise.

All of the above can be applied to the Omnideck, but it could be especially useful for bettering human health. It could be useful for rehabilitation training and simulations for people recovering from injuries or with difficulty walking. Even for gaming, it is healthier to exercise by physically walking around rather than sitting in a chair. With the research done on the Omnideck, future applications, games and use cases will be more accurate and provide better results which improves the viability of this system to be used sustainably.

References

A.G.E Tech. (1989). *Powerglove*. Retrieved May 2023, from ageinc: https://web.archive.org/web/20150525222703/http://www.ageinc.com/tech/index.html

Aaru Entertainment. (2022). *game-menu*. (Aaru Entertainment, LLC) Retrieved May 2023, from Aaru games: https://www.aaru.games/vrcade#game-menu

Alexander, A., Brunyé, T. T., Weil, S. A., & Weil, S. A. (2005, January). *From Gaming to Training: A Review of Studies on Fidelity, Immersion, Presence, and Buy-in and Their Effects on Transfer in PC-Based Simulations and Games.* Woburn: Aptima, Inc. Retrieved May 2023, from https://www.researchgate.net/publication/228374028_From_Gaming_to_Training_A_Review_of_Studies_on_Fidelity_Immersion_Presence_and_Buy-in_and_Their_Effects_on_Transfer_in_PC-Based_Simulations_and_Games

Armstrong, M. (2023, February 28). *VR Gaming Destined to Remain Niche?* Retrieved May 2023, from Statista: https://www.statista.com/chart/29401/vr-gaming-penetration-rate-vs-video-game-penetration-global/

Army Technology. (n.d.). *Omnifinity launches Omnideck 6*. Retrieved May 2023, from army-technology: https://www.army-technology.com/videos/3921400777001/

Beck, K., Grenning, J., Martin, R. C., Beedle, M., Highsmith, J., Mellor, S., . . . Marick, B. (2001). *Manifesto for Agile Software Development*. Retrieved May 2023, from agilemanifesto: http://agilemanifesto.org/

Beep! Medadrive. (1994, July). *News & Information.* Retrieved May 2023, from Segaretro: https://segaretro.org/index.php?title=File:BeepMD_JP_1994-08.pdf&page=31

Bond, M., & Beale, R. (2009). What makes a good game?: using reviews to inform design. *BCS-HCI* (p. 6). Cambridge: the British Computer Society. doi:10.5555/1671011.1671065

Computer Gaming World. (1986, December). Retrieved May 2023, from Computer Gaming World Museum: https://www.cgwmuseum.org/galleries/index.php?year=1986&pub=2&id=33

Crawford, C. (1984). *The Art of Computer Game Design.* Berkeley, California, USA: McGraw-Hill/Osborne Media. Retrieved May 2023

Csikszentmihalyi, M. (1975). *Beyond Boredom and Anxiety.* Michigan: Jossey-Bass Publishers. Retrieved May 2023

Davis, S., Nesbitt, K., & Nalivaiko, E. (2014). A Systematic Review of Cybersickness. *2014 Conference on Interactive Entertainment* (pp. 1-9). Association for Computing Machinery. doi:10.1145/2677758.2677780

De Angelo, J. (2000, June 10). *The Link Flight Trainer.* Retrieved May 2023, from ASME: https://www.asme.org/wwwasmeorg/media/resourcefiles/aboutasme/who%20we%20are/engineering%20history/landmarks/210-link-c-3-flight-trainer.pdf

Ellis, M. (n.d.). *The spectacle of the panorama*. Retrieved May 2023, from British Library: https://www.bl.uk/picturing-places/articles/the-spectacle-of-the-panorama|

Grau, O. (2002). *Virtual Art: From Illusion to Immersion.* The MIT Press. doi:10.7551/mitpress/7104.001.0001

Harding-Rolls, P. (2022, July 5). *Global games market forecast to decline in 2022*. (Ampere Analysis) Retrieved May 2023, from Ampere Analysis: https://www.ampereanalysis.com/insight/global-games-market-forecast-to-decline-in-2022

Heilig, M. L. (1960, October 4). *USA Patent No. 2,955,156.* Retrieved May 2023, from https://patents.google.com/patent/US2955156A/en

Heilig, M. L. (1962, August 28). *USA Patent No. 3,050,870.* Retrieved May 2023, from https://patentimages.storage.googleapis.com/90/34/2f/24615bb97ad68e/US3050870.pdf

Hendrix, C., & Barfield, W. (1996, August 1). Presence within Virtual Environments as a Function of Visual Display Parameters. *Presence: Teleoperators and Virtual Environments, 5*(3), pp. 274-289. doi:10.1162/pres.1996.5.3.274

Hendrix, C., & Barfield, W. (1996, August 1). The Sense of Presence within Auditory Virtual Environments. *Presence: Teleoperators and Virtual Environments, 5*(3), pp. 290-301. doi:10.1162/pres.1996.5.3.290

Howarth, J. (2023, January 18). *How Many Gamers Are There? (New 2023 Statistics)*. Retrieved May 2023, from Exploding Topics: https://explodingtopics.com/blog/number-of-gamers#editor-picks

IEA. (2023). *Global Energy Crisis*. Retrieved May 2023, from IEA: https://www.iea.org/topics/global-energy-crisis

Jerald, J. (2015). *The VR Book: Human-Centered Design for Virtual Reality.* (M. T. Özsu, Ed.) Association for Computing Machinery and Morgan & Claypool. doi:10.1145/2792790

Kassuba, T., Klinge, C., Hölig, C., Röder, B., & Siebner, H. R. (2013, January 15). Vision holds a greater share in visuo-haptic object recognition than touch. *NeuroImage, 65*, pp. 59-68. doi:10.1016/j.neuroimage.2012.09.054

Kedmey, D. (2015, June 11). *Here's When You Can Buy Oculus's Long-Awaited Virtual Reality Headset*. Retrieved May 2023, from Time: https://time.com/3918081/oculus-rift-release-date-microsoft/

Khimeche, L., & Wolf, S. (2017). *Mixed Reality, the Disruptive Technology to Increase Exercise Realism, Improve Live Training Efficiency and Shorten Exercise Preparation .* North Atlantic Treaty Organization Science and Technology Organization. Retrieved May 2023, from https://www.sto.nato.int/publications/STO%20Meeting%20Proceedings/Forms/All%20MPs.aspx?View=%7B72ed425f-c31f-451c-a545-41122bba61a7%7D&RootFolder=%2Fpublications%2FSTO%20Meeting%20Proceedings%2FSTO%2DMP%2DMSG%2D149&TreeField=Folders&TreeValue=STO%2DMP%2DM

Kickstarter. (2012, August 1). *Oculus Rift: Step Into the Game*. Retrieved May 2023, from Kickstarter: https://www.kickstarter.com/projects/1523379957/oculus-rift-step-into-the-game

Louis, T., Troccaz, J., Rochet-Capellan, A., & Bérard, F. (2019). Is it Real? Measuring the Effect of Resolution, Latency, Frame rate and Jitter on the Presence of Virtual Entities. *2019 ACM International Conference on Interactive Surfaces and Spaces* (pp. 5-16). New York: Association for Computing Machinery. doi:10.1145/3343055.3359710

Luckey, P. (2022, April 15). *10 Years of Oculus*. Retrieved May 2023, from Palmerluckey: https://palmerluckey.com/10-years-of-oculus/

Markoff, J. (1994, November 14). Nintendo Counts on a New 'Virtual' Game. San Fransisco, California, USA: The New York Times. Retrieved MAy 2023, from https://www.nytimes.com/1994/11/14/business/nintendo-counts-on-a-new-virtual-game.html

Mayor, J., Raya, L., & Sanchez, A. (2019, May 7). A comparative study of virtual reality methods of. *IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING, 9*(3). doi:10.1109/TETC.2019.2915287

McMahan, R. P., Bowman, D. A., Zielinski, D. J., & Brady, R. B. (2012, April 09). Evaluating Display Fidelity and Interaction Fidelity. *IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, 18*(4), p. 8. doi:10.1109/TVCG.2012.43

Merriam-Webster. (2023, May 31). *fidelity*. (Merriam-Webster) Retrieved May 31, 2023, from Merriam-Webster Dictionary: https://www.merriam-webster.com/dictionary/fidelity

Merriam-Webster. (2023, June 10). *limit*. Retrieved June 10, 2023, from Merriam-Webster.com Dictionary: https://www.merriam-webster.com/dictionary/limit

Merriam-Webster Dictionary. (2023, April 15). *scientific method*. Retrieved May 2023, from Merriam-Webster Dictionary: https://www.merriam-webster.com/dictionary/scientific%20method

Nabiyouni, M. (2017). *How Does Interaction Fidelity Influence.* Virginia Polytechnic Institute, Blacksburg. Retrieved May 2023, from https://vtechworks.lib.vt.edu/bitstream/handle/10919/74945/Nabiyouni_M_D_2017_support_1.pdf

Nystrom, R. (2014). Component. In R. Nystrom, *Game Programming Patterns* (p. 354). Genever Benning. Retrieved April 2023, from https://gameprogrammingpatterns.com/

Oman, C. M. (1990, February). Motion sickness: a synthesis and evaluation of the sensory conflict theory. *Canadian Journal of Physiology and Pharmacology, 68*, 294-303. doi:10.1139/y90-044

Omnifinity. (n.d.). *Omnifinity*. (Omnifinity) Retrieved June 2023, from Omnifinity: https://www.omnifinity.se/

Petry, N. M., Rehbein, F., Gentile, D. A., Lemmens, J. S., Rumpf, H.-J., Mößle, T., . . . O'Brien, C. P. (2014, January 23). An international consensus for assessing internet gaming disorder using the new DSM-5 approach. *Addiction, 109*(9), pp. 1399-1406. doi:10.1111/add.12457

Program-Ace. (2023, February 2). *Unity vs. Unreal: What to Choose for Your Project?* Retrieved May 2023, from Program-Ace: https://program-ace.com/blog/unity-vs-unreal/

Retro Game Mechanics Explained. (2019, July 13). Pac-Man Ghost AI Explained. Retrieved March 2023, from https://www.youtube.com/watch?v=ataGotQ7ir8

Schuemie, M. J., Van der Straaten, P., Krijn, M., & Van der Mast, C. A. (2001, May). Research on Presence in Virtual Reality: A Survey. *CYBERPSYCHOLOGY & BEHAVIOR, 4*, p. 19. doi:10.1089/109493101300117884

Sells, S. B., & Fixott, R. S. (1957, August). Evaluation of Research on Effects of Visual Training on Visual Functions. *American Journal of Ophthalmology, 44*(2), pp. 230-236. doi:10.1016/0002-9394(57)90012-0

Slater, M., & Wilbur, S. (1997, December 1). A framework for immersive virtual environments five: Speculations on the role of presence in virtual environments. *Presence: Teleoperators and Virtual Environments, 6*(6), pp. 603-616. doi:10.1162/pres.1997.6.6.603

Stenros, J. (2016, July). The Game Definition. *Games and Culture, 12*(6), p. 22. doi:10.1177/1555412016655679

Sutherland, I. E. (1965). *The Ultimate Display.* Information Processing Techniques Office, ARPA, OSD. Retrieved May 2023, from http://worrydream.com/refs/Sutherland%20-%20The%20Ultimate%20Display.pdf

Thor, P., & Omnifinity. (2023, January 18). *Omnifinity/Unity-SteamVR-API*. Retrieved May 2023, from GitHub: https://github.com/Omnifinity/Unity-SteamVR-API

United Nations. (2016). *Ensure access to affordable, reliable, sustainable and modern energy for all*. Retrieved May 2023, from sdgs.un: https://sdgs.un.org/goals/goal7

United Nations. (2016). *Promote peaceful and inclusive societies for sustainable development, provide access to justice for all and build effective, accountable and inclusive institutions at all levels*. Retrieved May 2023, from sdgs.un: https://sdgs.un.org/goals/goal16

United Nations. (2016). *Reduce inequality within and among countries*. Retrieved May 2023, from sdgs.un: https://sdgs.un.org/goals/goal10

Unity. (n.d.). *Entity Component System*. Retrieved April 2023, from Unity Manual: https://docs.unity3d.com/Packages/com.unity.entities@0.1/manual/index.html

Vaishnavi, V., & Kuechler, B. (2004, January 20). *Design Science Research in Information Systems.* Retrieved May 2023, from Association for Information Systems: http://www.desrist.org/design-research-in-information-systems/

Weech, S., Kenny, S., & Barnett-Cowan, M. (2019, February 4). Presence and Cybersickness in Virtual Reality Are Negatively Related: A Review. *Frontiers in Psychology, 10*. doi:10.3389/fpsyg.2019.00158

Wheatstone, C. (1838, June 21). *Contributions to the Physiology of Vision.—Part the First. On some remarkable, and hitherto unobserved, Phenomena of Binocular Vision. By CHARLES WHEATSTONE, F.R.S., Professor of Experimental Philosophy in King's College, London.* Retrieved May 2023, from Stereoscopy: https://www.stereoscopy.com/library/wheatstone-paper1838.html

Williams, R. (2016, January 11). *HTC Vive pre-orders to start on February 29*. Retrieved May 2023, from The Telegraph: https://www.telegraph.co.uk/technology/news/12092607/HTC-Vive-pre-orders-to-start-on-February-29.html

Zimmerman, T. G. (1982, September 17). *USA Patent No. 4,542,291.* Retrieved May 2023, from https://patents.google.com/patent/US4542291A/en

Attachments

Pre-Project Plan

Pac-Man in VR on the Omnideck
Forprosjektplan

Versjon <1.2>

Revisjonshistorie

Dato	Versjon	Beskrivelse	Forfatter
12/03/2023-16/03/2023	1.0	Førsteutkast	Leonard Sandløkk Schiller
02/05/2023	1.1	Endring av problemstilling	Leonard Sandløkk Schiller
11/06/2023	1.2	Finjustering før levering	Leonard Sandløkk Schiller

Innholdsfortegnelse

Innholdsfortegnelse .... - 2 -
1. Mål og rammer .... - 3 -
1.1 Orientering .... - 3 -
1.2 Problemstilling / prosjektbeskrivelse og resultatmål .... - 3 -
1.3 Mål .... - 3 -
1.4 Rammer .... - 4 -
2. Organisering .... - 5 -
3. Gjennomføring .... - 5 -
3.1. Hovedaktiviteter .... - 5 -
3.2. Milepæler .... - 7 -
4. Oppfølging og kvalitetssikring .... - 7 -
4.1 Kvalitetssikring. .... - 7 -
4.2 Rapportering .... - 8 -
5. Risikovurdering .... - 8 -

1. Mål og rammer

1.1 Orientering.

Denne oppgaven utføres av Leonard Sandløkk Schiller som avsluttende bachelorprosjekt i ingeniørfag innenfor data ved NTNU Trondheim. Prosjektet er basert på oppgaven gitt av Alexander Holt ved NTNU om "PacMan og/eller Frogger i VR". Formålet med utviklingsdelen av oppgaven er å lage et VR-spill som bruker Omnidecken for bevegelse i spillet. Denne tredemølla er i VR Labben på Gløshaugen. I rapporten ønsker jeg å lage retningslinjer og forslag til videre spillutvikling i et lignende VR miljø med Omnideck som bevegelseskontroller. Disse vurderingene vil komme gjennom arbeidet og funnene gjort i løpet av arbeidsprosessen.

Jeg valgte denne oppgaven fordi jeg har interesse og tidligere erfaring med både VR og spillutvikling.

1.2 Problemstilling / prosjektbeskrivelse og resultatmål

I dette prosjektet skal jeg drøfte problemstillingen

"Hvilke begrensninger og problemer kan oppstå og burde tas hensyn til ved utvikling av spill I et VR-miljø med Omnideck som bevegelseskontroller?"

Jeg ønsker å drøfte potensielle begrensninger som kan oppstå når man bruker Omnideck som bevegelseskontroller i VR spill.

Oppgavens essens er å skape god veiledning for fremtidige utviklere som lager applikasjoner i et lignende miljø.

RQ 1:

Hvilke begrensninger finnes med Omnideck?

RQ2:

Hvordan kan disse begrensningene påvirke brukeren?

RQ3:

Kan begrensningene overkommes?

1.3 Mål

Resultatmål:

Interne:

Oppnå god karakter (A/B)

Lage en god prosjektrapport og ha en god sluttpresentasjon

Lage et bra og morsomt spill

Eksterne:

Utvikle god sluttrapport

Utvikle et bra spill

Effektmål:

Interne:

Arbeide regelmessig

Bli god på loggføring av prosess

Bli god på tidsfordeling mellom arbeidsoppgaver

Eksterne:

Drøfte begrensninger som finnes med Omnideck

Hensikten med prosjektet er å lage en god rapport som kan bli brukt til å vurdere valg ved utvikling av spill i VR og som kan bli brukt som veiledning.

Jeg ønsker å lage en morsom versjon av PacMan i VR med god spillopplevelsen for sluttbrukeren. Dette skal lages i Unity og lagres med versjonskontrollsystem som Git. Man skal kunne bevege seg ved å gå på Omnideck tredemølla.

Med dette prosjektet ønsker jeg også å bli god til å håndtere tid, frister og jobbe på en god måte for å ferdigstille sluttproduktet. Til slutt ønsker jeg å oppnå en god karakter (A/B).

1.4 Rammer

Vanlig behov for denne oppgaven inkluderer tilgang til PC og utviklingsprogrammet unity. Disse har gruppemedlemmene tilgang til på egenhånd. Det er ikke forutsatt behov for penger for denne oppgaven.

Denne oppgaven har flere spesielle behov både med henhold til utstyr og tid.

For denne oppgaven så er det spesifisert at man skal lage et spill i VR med bruk av en spesiell tredemølle som kan brukes med VR. Både tredemølla og VR utstyre finnes på VRLabben på NTNU gløshaugen og derfor er det nødvendig å ha adgang for både testing og bruk av disse systemene.

I tillegg er det spesielt behov for utsatt tidsramme for hele oppgaven ettersom gruppemedlemmene har vært på utveksling i Tyskland. Det Tyske semesteret er forskjøvet i forhold til det Norske og starter i oktober og ender i februar. På grunn av denne forskyvingen er det nødvendig med utsettelse av oppgavene og arbeidskrav med flere uker.

2. Organisering

Aktørene involvert i denne oppgaven består av oppdragsgiver, veileder og arbeidstaker. Denne oppgaven har kun en arbeidstaker, Leonard Sandløkk Schiller. Oppdragsgiver og veileder er samme person, Alexander Holt.

3. Gjennomføring

3.1. Hovedaktiviteter.

Hoved aktiviteter inkluderer:

Arbeidskrav

Spillutvikling

Dokumentasjon

Annet

3.1.1 Begrunnelse

Arbeidskravene må gjennomføres for å oppfylle alle kriteriene og oppgavene som kommer med bacheloroppgaven. Dette inkluderer forprosjektsplan, visjonsdokument, poster og presentasjon, hovedrapporten og til slutt presentasjon av oppgaven.

Spillet er en essensiell del av oppgaven og å ha et ferdigutviklet spill som resultat er nødvendig.

Dokumentasjon blir karaktersatt og er en viktig del av oppgaven, samtidig er det viktig å loggføre fremskritt over tid for å se hvordan utviklingen har skjedd gjennom hele prosjektet.

3.1.2 Forutsetninger

For at en gitt arbeidskravet kan fullføres må alle tidligere arbeidskrav fullføres.

For å utvikle et VR spill med tredemølle er det forutsatt at det er mulig å ha tilgang til VR utstyr inkludert tredemølle som skal brukes med spillet. Dette er strengt nødvendig for testing. I tillegg er det forutsatt tilgang til spillutviklingsverktøy som programmet unity, som er åpen og tilgjengelig for alle.

Dokumentasjon forutsetter at man arbeider med ting som må dokumenteres.

3.1.3 Gjennomføring

Alle hovedaktiviteter blir alle gjort av studenten.

Arbeidskravene vil bli gjort for det meste sekvensielt. For at en gitt arbeidskravet kan fullføres må alle tidligere arbeidskrav fullføres. Arbeidskrav en må være ferdig før arbeidskrav to kan gjøres, to før tre og tre gjøres før fire på grunn av tidsbegrensninger. Arbeidskrav 5 kan gjøres parallelt med arbeidskrav 4, men Arbeidskrav 4 må være ferdig før arbeidskrav 5 kan ferdigstilles. Arbeidskrav en skal gjøres frem til fredag 17 mars 2023. Arbeidskrav to og tre skal gjøres før 27 mars 2023. Arbeidskrav 4 og 5 skal gjøres ferdig til 12 Juni.

Spillutviklingsprosessen vil være sprintbasert og forsøke å etterligne en smidig utviklingsprosess. En fullstendig smidig utviklingsprosess med scrum-mester og møter er ikke mulig ettersom hele utviklingsprosessen blir kun gjennomført av en student. Derfor er fokuset og etterligningen av smidig utviklingsmetodologi på selve utviklingsprosessen med flere sprints og revurdering av muligheter, istedenfor å ha fokus på det gruppeorienterte. Spillutviklingen må gjennomføres parallelt med Arbeidskrav og dokumentasjon. Spillet må være lagret på versjonskontroll verktøy som git enten gjennom github eller gitlab. Utvikling av dette spillet skal skje sammen med veiledning av veileder og innspill fra oppdragsgiver slik at ferdig produktet blir som etterspurt.

Dokumentasjon skjer parallelt med alt annen arbeid. Dokumenter skal følge de gitte malene i blackboard og alle dokumenter nevnt i resultater skal ferdigstilles. Dokumentene varierer, og noen skal gjøres fortløpende, andre har spesifikke tidsfrister og noen må kun arbeides med før prosjektet leveres inn.

Spillutviklingen må skje før rapportskrivingen ettersom spillet skal brukes til forskning som skal skrives om i rapporten.

3.1.4 Resultater

Resultatet av arbeidskrav en skal gi en ferdig forprosjektsplan og visjonsdokument. Resultatet av arbeidskrav to skal gi en ferdig poster som kan presenteres i arbeidskrav tre. Resultatet av arbeidskrav tre er en gjennomført presentasjon av posteren. Resultatet av arbeidskrav fire er en ferdig hovedrapport. Resultatet av arbeidskrav fem er en fullført presentasjon av oppgaven.

Resultat av spillutviklingen skal være et ferdig og fungerende VR spill som oppfyller krav av oppdragsgiver.

Resultat av dokumentasjon er en veldokumentert oppgave og flere vedlegg inkludert:

Forprosjektsplan, visjonsdokument, arbeidskontrakt, møteinnkallinger, møtereferater, kravdokumentasjon, systemdokumentasjon, poster, presentasjon, prosesshåndbok, timeliste/arbeidslogg, detaljert beskrivelse, hovedrapport, evt. Annet.

Det kan oppstå flere arbeidsoppgaver ila. oppgaven.

3.2. Milepæler.

17.03.2023 Innlevering av Forprosjektplan og Visjonsdokument

27.03.2023 Innlevering av poster/plakat

29.03.2023 Presentasjon av poster/plakat

30.03.2023 Fremvisning av prototype av spillet

01.05.2023 Spillet ferdig utviklet

26.05.2023 Presentasjon av oppgaven

12.06.2023 Hovedrapport leveres inkludert dokumentasjon av arbeidsprosessen. Skal inkludere statusrapporter, møteinnkallinger, møtereferater og timelister innleveres

4. Oppfølging og kvalitetssikring

4.1 Kvalitetssikring.

For å sikre kvalitet på arbeidet vil vi ha møter og diskutere arbeidet som er gjort. I tillegg vil det bli gjort revisjoner på tidligere dokumenter for å forsikre kvalitet på arbeidet.

4.2 Rapportering.

Rapportering skal skje hver tirsdag de første 4 ukene og deretter annenhver tirsdag.

5. Risikovurdering

Tabell 1: Her illustreres risikoanalysen I form av en risikomatrise. Konsekvens og sannsynlighet er på en skala fra 1-5 or risikofaktor er konsekvensen ganger sannsynligheten på en skala fra 1-25

Risikobeskrivelse	Konsekvens/alvorlighet	Sannsynlighet	Risikofaktor	Tiltak
Forsinkelse	4	5	20	Denne oppgaven ble forsinket som følge av utvekslingssemesteret mitt I tyskland. Det tyske semesteret avslutter I Februar og jeg fikk ikke startet på oppgaven før Mars. Tiltaket tatt her er en utvidelse av tidsfristen for oppgave 4 og 5 med 3 uker.
Sykdom	5	2	10	Jeg kan bli syk og dette vil forhindre fremgang med oppgaven, det er spesiellt økt risiko ettersom oppgaven allerede er forsinket og jeg er alene på gruppa. Tiltaket er å prøve å holde meg frisk og dersom jeg blir syk prøve å bli frisk så fort som mulig.
VR Sykdom	2	4	8	Et viktig problem innenfor VR er VR sykdom. Dette kan sammenlignes med sjøsykdom der man kan for eksempel oppleve svimmelhet, kvalmhet dårlig orientasjon. Det er viktig å prøve å begrense dette så mye som mulig for både megselv og eventuellt frivillige testere.
Lav motivasjon	3	2	6	Jeg har slitt med motivasjon tidligere, men denne oppgaven er interessant for meg og det er en lavere risiko for lav motivasjon. Dersom jeg har lav motivasjon er det viktig å sette daglige mål slik at oppgaver blir gjort.
Dokument Tap	5	1	5	Tap av dokumenter er veldig alvorlig. Tiltak for dette er å laste ned kopier av dokumenter og bruke skytjenesten til microsoft.
Trademark problemer	?	?	?	PacMan er varemerket og konsekvensene kan være alvorlige. Det må tas en avgjørelse I hvor stor grad spillet kan etterlignes.
Andre Spillrelaterte risoker	?	?	?	Utover VR sykdom kan det oppstå andre problemer under bruk av VR utstyr. Disse skal forsøkes å forutses og forhindres.

Project Handbook

Prosjekthåndbok

Innhold

Framdriftsplan – Mileperlplan 1
Møteinnkallinger 2
Møtereferater 13
Timelister 28

Framdriftsplan – Mileperlplan

17.03.2023 Innlevering av Forprosjektplan og Visjonsdokument

27.03.2023 Innlevering av poster/plakat

29.03.2023 Presentasjon av poster/plakat

30.03.2023 Fremvisning av prototype av spillet

01.05.2023 Spillet ferdig utviklet

26.05.2023 Presentasjon av oppgaven

12.06.2023 Hovedrapport leveres inkludert dokumentasjon av arbeidsprosessen. Skal inkludere statusrapporter, møteinnkallinger, møtereferater og timelister innleveres

Møteinnkallinger

Møteinnkalling 1

Hei,

er litt sent ute pga. sykdom og en overraskende travel start på semesteret, men det ble bare slik.

Vi skulle gjerne ha tatt et lite møte, slik at vi kunne ha diskutert hvordan vi skal gjøre det fremover, gitt at du kommer til å være i utlandet en god stund ennå. Jeg er ikke helt sikker på hva som kommer til å fungere best, men kanskje du har gjort deg opp noen tanker om dette siden du sendte mailen. Det er jo avhengig av timeplanen der du er nå, også.

Har du tid til et møte på tirsdag klokka 11:30 i neste uke? Vi kan ta det på https://meet.idi.ntnu.no/frogger.

-- Sander

Møteinnkalling 2

Innkalling til møte: Bacheloroppgave 90
Tidspunkt/sted: 07.03.23 kl 12:00-12:30, https://meet.idi.ntnu.no/frogger
Følgende personer innkalles:
Studenter: Leonard Sandløkk
Oppdragsgiver: Alexander Holt
Agenda:
Sak 01/23 Godkjenning av møteinnkalling
Sak 02/23 Referat av forrige møte gjennomgås
Sak 03/23 Diskusjon om tidsfrister og arbeidskrav
Sak 04/23 Diskusjon/veiledning mht. problemstilling og forprosjektplan
Sak 05/23 Videre arbeid
Sak 06/23 Eventuelle saker
Møtet planlegges avsluttet 12:30 men kan ta mer tid. Ta kontakt med undertegnede dersom du ikke har anledning til å møte opp eller ønsker å endre tidspunkt.
Mvh,
Leonard Lørenskog 06.03.23

Møteinnkalling 3

Innkalling til møte: Bacheloroppgave 90
Tidspunkt/sted: 14.03.23 kl 15:00-15:30,
rom 201, IT-syd
Følgende personer innkalles:
Studenter: Leonard Sandløkk
Oppdragsgiver: Alexander Holt
Agenda:
Sak 01/23
Sak 02/23 Referat av forrige møte gjennomgås
Sak 03/23 Addresse/kontaktinformasjon
Sak 04/23 Arbeidskontrakt/3 parts avtale
Sak 05/23 Rapportering
Sak 06/23 Diskusjon av kritiske datoer
Sak 07/23 Produkt visjon/krav/mål
Sak 08/23 CI/CD
Sak 09/23 Videre arbeid
Sak 10/23 Eventuelle saker

Møtet planlegges avsluttet 15:30 men kan ta mer tid. Ta kontakt med undertegnede dersom du ikke har anledning til å møte opp eller ønsker å endre tidspunkt.
Mvh,
Leonard Trondheim 13.03.23

Møteinnkalling 4

Innkalling til møte: Bacheloroppgave 90
Tidspunkt/sted: 21.03.23 kl 11:30-12:00, rom 201, IT-syd
Følgende personer innkalles:
Studenter: Leonard Sandløkk
Oppdragsgiver: Alexander Holt
Agenda:
Sak 01/23 Godkjenning av møteinnkalling
Sak 02/23 Referat av forrige møte gjennomgås
Sak 03/23 VizLab booking (og tour?)
Sak 04/23 Poster/plakat
Sak 05/23 gjennomgåing av leverte arbeidskrav (Forprosjektsplan og visjonsdokument)
Sak 06/23 Problemstilling og forskningsspørsmål
Sak 07/23 Videre arbeid
Sak 08/23 Eventuelle saker
Møtet planlegges avsluttet 11:30 men kan ta mer tid. Ta kontakt med undertegnede dersom du ikke har anledning til å møte opp eller ønsker å endre tidspunkt.
Mvh,
Leonard Trondheim 21.03.23

Møteinnkalling 5

Innkalling til møte: Bacheloroppgave 90
Tidspunkt/sted: 30.03.23 kl 13:30-14:00, rom 201, IT-syd
Følgende personer innkalles:
Studenter: Leonard Sandløkk
Oppdragsgiver: Alexander Holt
Agenda:
Sak 01/23 Godkjenning av møteinnkalling
Sak 02/23 Referat av forrige møte gjennomgås
Sak 03/23 Diskusjon av gjennomført presentasjon
Sak 04/23 Diskusjon om prototype
Sak 05/23 VRLab?
Sak 06/23 Videre arbeid
Sak 07/23 Eventuelle saker
Møtet planlegges avsluttet 14:00 men kan ta mer tid. Ta kontakt med undertegnede dersom du ikke har anledning til å møte opp eller ønsker å endre tidspunkt.
Mvh,
Leonard Trondheim 30.03.23

Møteinnkalling 6

Innkalling til møte: Bacheloroppgave 90
Tidspunkt/sted: 13.04.23 kl 16:00-17:00, VR-Lab, IT-syd
Følgende personer innkalles:
Studenter: Leonard Sandløkk
Oppdragsgiver: Alexander Holt
Agenda:
Sak 01/23 Godkjenning av møteinnkalling
Sak 02/23 Referat av forrige møte gjennomgås
Sak 03/23 Diskusjon om prototype
Sak 04/23 Diskusjon av videre arbeid på prototype
Sak 05/23 Diskusjon av videre planlegging
Sak 06/23 Videre arbeid
Sak 07/23 Eventuelle saker
Møtet kan ta mer tid ettersom første prototype av spillet skal testes. Ta kontakt med undertegnede dersom du ikke har anledning til å møte opp eller ønsker å endre tidspunkt.
Mvh,
Leonard Trondheim 13.04.23

Møteinnkalling 7

Innkalling til møte: Bacheloroppgave 90
Tidspunkt/sted: 02.05.23 kl 14:30-15:30, VRLabben
Følgende personer innkalles:
Studenter: Leonard Sandløkk
Oppdragsgiver: Alexander Holt
Agenda:
Sak 01/23 Godkjenning av møteinnkalling
Sak 02/23 Referat av forrige møte gjennomgås
Sak 03/23 Vurdering av prototype
Sak 04/23 Videre arbeid
Sak 05/23 Eventuelle saker

Møtet planlegges avsluttet 15:30 men kan ta mer tid. Ta kontakt med undertegnede dersom du ikke har anledning til å møte opp eller ønsker å endre tidspunkt.
Mvh,
Leonard Trondheim 02.05.23

Møteinnkalling 8

Innkalling til møte: Bacheloroppgave 90
Tidspunkt/sted: 16.05.23 kl 11:00-12:00, rom 201 itbygget
Følgende personer innkalles:
Studenter: Leonard Sandløkk
Oppdragsgiver: Alexander Holt
Agenda:
Sak 01/23 Godkjenning av møteinnkalling
Sak 02/23 Referat av forrige møte gjennomgås
Sak 03/23 Gjennomgåelse av eposter/spørsmål
Sak 04/23 Gjennomgåelse av statusrapporter
Sak 05/23 Hovedrapport
Sak 06/23 Videre arbeid
Sak 07/23 Eventuelle saker

Møtet planlegges avsluttet 12:00 men kan ta mer tid. Ta kontakt med undertegnede dersom du ikke har anledning til å møte opp eller ønsker å endre tidspunkt.
Mvh,
Leonard Trondheim 15.05.23

Møteinnkalling 9

Innkalling til møte: Bacheloroppgave 90
Tidspunkt/sted: 26.05.23 kl 14:00-15:00, rom 201 itbygget
Følgende personer innkalles:
Studenter: Leonard Sandløkk
Oppdragsgiver: Alexander Holt
Agenda:
Sak 01/23 Godkjenning av møteinnkalling
Sak 02/23 Referat av forrige møte gjennomgås
Sak 03/23 Gjennomgåelse av statusrapporter
Sak 04/23 Hovedrapport
Sak 06/23 Videre arbeid
Sak 07/23 Eventuelle saker

Møtet planlegges avsluttet 15:00 men kan ta mindre tid. Ta kontakt med undertegnede dersom du ikke har anledning til å møte opp eller ønsker å endre tidspunkt.
Mvh,
Leonard   Trondheim 25.05.23

Møteinnkalling 10

Innkalling til møte: Bacheloroppgave 90
Tidspunkt/sted: 08.06.23 kl 11:00-12:00, https://meet.idi.ntnu.no/leonard
Følgende personer innkalles:
Studenter: Leonard Sandløkk
Oppdragsgiver: Alexander Holt
Agenda:
Sak 01/23 Godkjenning av møteinnkalling
Sak 02/23 Referat av forrige møte gjennomgås
Sak 03/23 Gjennomgåelse av statusrapporter
Sak 04/23 Hovedrapport
Sak 05/23 Videre arbeid
Sak 06/23 Eventuelle saker

Møtet planlegges avsluttet 12:00 men kan ta mindre tid. Ta kontakt med undertegnede dersom du ikke har anledning til å møte opp eller ønsker å endre tidspunkt.
Mvh,
Leonard   Trondheim 07.06.23

Møtereferater

Møtereferat 1

Referat fra prosjektmøte bacheloroppgave 90
Dato og tid:25.01.23 kl 13:00 - 14:00
Sted: https://meet.idi.ntnu.no/frogger
Til stede: Leonard Sandløkk, Alexander Holt
Frafall: Ingen
Møteleder: Alexander Holt

Sak 01/2023
Første møte, ingen forrige møtereferat å gjennomgå.

Sak 02/2023
Diskuterte om det var mulighet å legge til en student i bachelorgruppa. Dette viste seg å være potensielt mulig, men kandidaten trakk seg og punktet ble deretter strøket.

Sak 03/2023
Det ble diskutert hvordan å utsette oppgaven med tanke på utveksling og det ble bestemt at Leonard skulle ta kontakt med Grethe. Etter oppklaring med Grethe så ble det bestemt at Alexander har siste avslutning angående tidsfrister og at antatt maksimal utsettelse er to til tre uker.

Sak 04/2023
Diskuterte hva som gjør en god rapport og hvilke fallgruver å unngå. Det ble lagt spesielt vekt på å ha en rød tråd gjennom hele oppgaven slik at den gir mening og har god flyt.

Sak 05/2023
Problemstilling for oppgaven ble diskutert, men ikke fastlagt. Forslag handlet om brukeropplevelser av Omnideck eller å utarbeide råd for utvikling av spill i samsvar med Omnideck.

Sak 06/2023
Valg av utviklingsprogram ble diskutert. Valgene sto hovedsakelig mellom Unity og Unreal Engine. Unity ble valgt på grunn av eksisterende erfaring med programmet.

Sak 07/2023
Utgangspunkt for oppgaven ble diskutert. Oppgaven er fremdeles til dels åpen, men det ble bestemt å snevre inn oppgaven til et retro spill som PacMan eller Frogger som allerede sto i oppgavens beskrivelse. Disse spillene var valgt på grunn av viktigheten av bevegelse i spillet.

Sak 08/2023
Ambisjonsnivået til oppgaven med tanke på karakter og spillets “scope” ble diskutert, men ble ikke bestemt på grunn av uklarheter som følge av usikre tidskrav.

Sak 09/2023

Vurderingskriterier for oppgaven ble oppklart, omtrent 40% av karakteren baseres på det ferdige spillet, 40% blir basert på hovedrapporten og 20% blir basert på prosessen. Disse tallene er omtrentlig og ment til bruk for veiledning.

Sak 10/2023
Det ble diskutert om hvordan fremgangsmåten og prosessen for en bacheloroppgave skal gjennomføres. Det legges mye vekt på om oppgaven er fullført på selvstendig vis.

Lørenskog 03.03.2023 Leonard

Møtereferat 2

Referat fra prosjektmøte bacheloroppgave 90

Dato og tid: 07.03.23 kl 12:00-12:30
Sted: https://meet.idi.ntnu.no/frogger
Til stede: Leonard Sandløkk, Alexander Holt
Frafall: Ingen
Møteleder: Leonard Sandløkk

Sak 01/23
Møteinnkalling godkjent.

Sak 02/23
Referat fra forrige møte gjennomgått og godkjent uten merknader.

Sak 03/23
Alexander godkjenner utsettelse av bacheloroppgaven, men må også godkjennes av Grethe og Ingrid for å bekrefte at det ikke vil skape for alvorlige administrative problemer.
Arbeidskrav 1 kan utsettes til Fredag 17 Mars 2023, men Arbeidskrav 2 og 3 kan ikke utsettes siden dette er en fysisk presentasjon med fastsatte dager. Arbeidskrav 1 må fullføres før Arbeidskrav 2 og 3.
Det er viktig å gå nøye gjennom malene og retningslinjene for å forstå kravene til oppgavene. Spesielt med hovedrapporten er dette viktig.
Arbeidskrav 2 og 3 handler om å lage en poster og presentere denne. Det var usikkerhet om nøyaktig hva som inngikk i disse to og det ble bestemt at jeg skulle sende epost til Grethe for oppklaring.
Det ble bestemt at Gant diagram ikke var optimalt og vi ønsker å bruke mileperlplan istedenfor.

Sak 04/23
Det er viktig å følge malene for å gjøre en god oppgave. Likevel kan det være grunnlag for avvik fra malene, men dette burde kun skje om det gir mening, og det funker ikke å tilfeldigvis fjerne deler fra malen.
For oppgaven er det viktig å tenke på mottakeren, nemlig sensor. Sensoren er ekstern og er ikke med gjennom prosessen av oppgaven, de får kun sluttproduktet. Dette er viktig å tenke på når man skriver slik at for eksempel innlysende ting gjennom prosessen også blir formulert som innlysende for sensoren.

Sak 05/23
Det er viktig å tenke på problemstilling og forskningsspørsmål til neste møte.

Sak 06/23 Eventuelle saker
Fysisk møte planlagt for 15:00 Tirsdag 14 Mars og neste faglig møte planlagt for 11:30 Tirsdag 21 Mars.

Lenke til VRLab påmeldingsliste oppgitt:
https://docs.google.com/spreadsheets/d/1mnE3emxrgcmQpXZgJl83oIF-cMgOQUe5DJZA8UOBS3M/edit#gid=513650056

Lørenskog 06.03.23 Leonard

Møtereferat 3

Referat fra prosjektmøte bacheloroppgave 90

Dato og tid: 14.03.23 kl 15:00-15:30
Sted: Rom 201, 2 etg. IT-syd
Til stede: Leonard Sandløkk, Alexander Holt
Frafall: Ingen
Møteleder: Leonard Sandløkk

Sak 01/23
Møteinnkalling godkjent.

Sak 02/23
Referat fra forrige møte gjennomgått og godkjent uten merknader.

Sak 03/23
Det ble bestemt at deling av adresse og kontaktinformasjon var unødvendig og e-post var tilstrekkelig.

Sak 04/23
Det ble bestemt at arbeidskontrakt og tre parts avtale var unødvendig ettersom oppgaven er gitt av NTNU og det er ingen andre på gruppa.

Sak 05/23
Det ble bestemt at rapportering skal skje ukentlig i startfasen og skal trappes ned til annenhver uke senere i oppgavens løp.

Sak 06/23
Det ble bestemt at innleveringsfristene og viktige datoene som sto på Blackboard skulle følges med unntak av forprosjekts plan og visjonsdokument. Det er usikkerhet når presentasjon av oppgaven skjer på grunn av utsettelse.
17.03.2023 Forprosjektplan og Visjonsdokument
27.03 2023 innlevering av poster/plakat
30.03.2023 presentasjon av poster/plakat
30.03.2023 prototype av spillet ønskes
12.06.2023 Dokumentasjon av arbeidsprosessen skal inkludere statusrapporter, møteinnkallinger, møtereferater og timelister
16.06.2023 presentasjon av oppgaven

Det er viktig å merke at krav 4, prototype av spillet, ble satt veldig tidlig. Dette var fordi planlagte datoen for neste møte 04.04.2023 var i påsken og kunne ikke ta sted. Derfor ble kravet satt før påsken med forventning at den kunne utsettes til etter påsken dersom nødvendig.

Sak 07/23

Hovedmålet for ferdigproduktet er å ha en spillbar versjon av PacMan i VR med bruk av Omnideck som bevegelseskontroller. Rammene for oppgaven er generelle og åpne, noen av de fastsatte kriteriene er å bruke både VR headset og Omnideck tredemølla på VRLabben. Utover dette ble det snakket om oppgavens "scope". Det ble enighet om å nedprioritere flerspiller ettersom det vil ikke bidra substansielt til problemstillingene og forskningsspørsmålene som ble diskutert.

Sak 08/23 CI/CD
Det ble bestemt at CI/CD var utenfor rammene av oppgaven og dermed unødvendig.

Sak 09/23 Videre arbeid
Det viktigste videre arbeidet er arbeid på plakat, poster og prototype av spillet.
Problemstilling ble fastsatt til:
"Hvordan bruk av omnideck som en bevegelseskontroller påvirker spillopplevelse i VR og hva er fornuftige valg ved utvikling av spill i et slikt miljø?"
Forskningsspørsmål ble foreslått i dette møtet, men konkrete forskningsspørsmål ble ikke bestemt ettersom det fremdeles er frihet i hvordan oppgaven løses.

Sak 10/23 Eventuelle saker
Det ble oppklart at det sannsynligvis er mulig å søke mastergrad i år.
Neste planlagt møte 11:30 Tirsdag 21 Mars.

Trondheim 16.03.23 Leonard

Møtereferat 4

Referat fra prosjektmøte bacheloroppgave 90

Dato og tid: 21.03.23 kl 11:30-12:00
Sted: Rom 201, 2 etg. IT-syd
Til stede: Leonard Sandløkk, Alexander Holt
Frafall: Ingen
Møteleder: Leonard Sandløkk

Sak 01/23
Møteinnkalling godkjent.

Sak 02/23
Referat fra forrige møte gjennomgått og godkjent uten merknader.

Sak 03/23
Alexander ga en omvisning av Vizlabben og forklarte bruk av Omnideck og bookingsystemet.

Sak 04/23
Utdypning av arbeidskravene ble gitt av Grethe og sto på Blackboard.

Sak 05/23
Forprosjekts plan og visjonsdokument ble ikke gjennomgått ettersom veileder ikke leste gjennom. Dette var godkjent ettersom de var utkast og skulle endres i noen grad i løpet av prosessen.

Sak 06/23
Problemstillingen ble vurdert igjen. Den er todelt og kan ha for store rammer spesielt med tanke på tidsbegrensningene og størrelsen på gruppa. Det kan hende denne blir snevret i fremtiden. Begge deler av problemstillingen fikk mer konkrete forskningsspørsmål, men det kan hende noen blir nedprioritert eller fjernet.

Sak 07/23
Det er viktig å få gjort ferdig plakat og poster til tidsfristen. I tillegg viktig å lage en spillbar prototype til enten 30.03.2023 eller en senere dato uken etter påske.

Sak 08/23 Eventuelle saker
Neste planlagt møte 13:30 Torsdag 30 Mars. Møtet foregår senere I uka for å gi mer tid til utvikling av prototype.

Trondheim 21.03.23 Leonard

Møtereferat 5

Referat fra prosjektmøte bacheloroppgave 90

Dato og tid: 30.03.23 kl 13:30-14:00
Sted: rom 201, IT-syd
Til stede: Leonard Sandløkk, Alexander Holt
Frafall: Ingen
Møteleder: Leonard Sandløkk

Sak 01/23
Møteinnkalling godkjent.

Sak 02/23
Referat fra forrige møte gjennomgått og godkjent uten merknader.

Sak 03/23
Dette møtet foregikk direkte etter presentasjonen. Det ble diskutert hvordan presentasjonen gikk og den ble godkjent med ingen merknader. Presentasjonen hadde både en poster og en powerpoint presentasjon grunnet en misforståelse, begge ble presentert og presentasjonen var under 10 minutter.

Sak 04/23
Det var lite diskusjon angående prototypen. Hovedsakelig hvor mye som hadde blitt ferdigstilt og om den var testbar. Prototypen besto av spillebanen, men hadde ikke hovedkarakteren eller bevegelse og kunne ikke testes.

Sak 05/23
Det var ikke en ferdigstilt spillbar versjon av prototypen og det ble ikke testet eller sett på i VRLabben. Dette var forventet grunnet kort tidsrom, annet arbeid, og dårlig tilgang til VRLabben grunnet arbeid fra masterstudenter.

Sak 06/23 Videre arbeid
Ferdigstille en spillbar versjon av prototypen i påsken for å teste etter påsken.

Sak 07/23 Eventuelle saker
Neste møtet skal skje første uka etter påskeferien, foreløpig ingen fastsatt dato eller tid.

Trondheim 30.03.23 Leonard

Møtereferat 6

Referat fra prosjektmøte bacheloroppgave 90

Dato og tid: 13.04.23 kl 16:00-17:00,
Sted: VR-Lab, -1 etg. IT-syd
Til stede: Leonard Sandløkk, Alexander Holt
Frafall: Ingen
Møteleder: Leonard Sandløkk

Sak 01/23
Møteinnkalling godkjent.

Sak 02/23
Referat fra forrige møte gjennomgått og godkjent uten merknader.

Sak 03/23
Prototypen ble testet og vurdert av Alexander.
Prototypen besto av et MVP. Spilleren kunne blant annet gå rundt i PacMan labyrinthen, spise pellets og power pellets, bli jagd av spøkelser og skremme dem og se sin poengsum. Dette var minimum nødvendige funksjonelle kravene for å etterligne PacMan.

Sak 04/23
Det ble diskutert om spillet trengte flere egenskaper. Jeg kom med forslag om å legge til gjenstander som kan brukes mot spøkelsene for å gjøre spillet mer morsomt og som funker bra med VR miljøet. Dette ble godkjent og utviklingsprioriteten til neste møte.

Sak 05/23
Det ble diskutert når spillet skulle stå ferdigstillt og hvor mye tid som ville stå igjen etter spillets utvikling. Det ble bestemt at spillet burde forsøkes å bli ferdigutviklet rundt slutten av April og begynnelsen av Mai for å ha tilstrekkelig tid for skriving og dokumentasjon. I tillegg ble det diskutert om det kunne gjennomføres brukertester, dersom disse skulle gjennomføres burde det være I begynnelsen av Mai.

Sak 06/23
Videre arbeid er å legge til gjenstander for spilleren.

Sak 07/23
Neste møtet planlagt Tirsdag 25 April, ingen tid avsatt enda.

Trondheim 13.04.23 Leonard

Møtereferat 7

Referat fra prosjektmøte bacheloroppgave 90

Dato og tid: 02.05.23 kl 14:30-15:30, VRLabben
Til stede: Leonard Sandløkk, Alexander Holt
Frafall: Ingen
Møteleder: Leonard Sandløkk

Sak 01/23
Møteinnkalling godkjent.

Sak 02/23
Referat fra forrige møte gjennomgått og godkjent uten merknader.

Sak 03/23
Prototypen ble testet og vurdert av Alexander.
Prototypen besto av det ferdige spillet og alle ønsket funksjoner. Fra forrige versjon ble det introdusert frukt som ble til gjenstander som kunne brukes av spilleren på forskjellige måter mot spøkelsene, ny spøkelses oppførsel som etterligner original PacMan, et kart, ny spiller modell, fungerende UI med menyvalg, instillinger og "highscore" tabell, teleportasjon, og lydavspilling. Dette var all ønsket ekstra og foreslått funksjonalitet og mer.
Spillet ble godkjent med forslag om å stryke ut noen få igjenstående bugs.
Alexander la merke til bevegelsen hans som kom av Omnidecken, spesifikt "zikk-zakk" gåing og potensiell problem med "dødsone" I midten av Omnidecken. Lignende opplevelser ble notert ned tidligere I mine personlige tester.

Sak 04/23
Ettersom spillet var ferdigstilt ble det foreslått å jobbe på skrivedelen og dokumentasjon delen av oppgaven I tillegg til å stryke ut de gjenstående "bugs"-ene som var igjen I spillet.
Oppgaveomfanget, brukertester og tidsbegrensningene ble diskutert. På grunn av dårlig tid ble det bestemt at det blir for risikabelt og tidskrevende å gjennomføre brukertester. På grunn av dette må den todelte oppgaven ta en ny vinkel uten fokus på brukeropplevelse og mer fokus på andre del av problemstillingen, angående bruk av Omnideck og forslag til spillutvikling I samsvar med Omnideck.
Med den nye vinkelen blir fokuset på mekanikken bak Omnidecken, og hvordan det påvirker spilleren mens de spiller, og hvordan ulempene kan minimeres. I tillegg ønskes retningslinjer for utvikling I samsvar med Omnideck.
Koden og systemet må dokumenteres.
Potensiellt lag rom som fremhever forvente "problemer" med Omnideck.

Sak 05/23
Dette møtet var forsinket. Spillet var I prinsippet ferdig før Tirsdag 25 April og var klar for testing da med unntak av planlagt bugfixing etter møtet. Dette møtet skjedde ikke grunnet en manglende epost-notifikasjon og måtte utsettes. Dette var uheldig ettersom videre arbeid kunne ikke gjøres ettersom spillet ikke var godkjent og videre arbeid på spillet var minimalt

ettersom den var hovedsaklig ferdig. Dette møtet skjedde ved første mulige anledningen etter 25 April.
Viktig å lisensiere brukte ressurser på riktig måte.
Planlagt møter Tirsdag 16 April og Tirsdag 30 April for statusrapporter på skriveprosessen.

Trondheim 08.05.23 Leonard

Møtereferat 8

Referat fra prosjektmøte bacheloroppgave 90
dato/tid: 16.05.23 kl 11:00-12:00, rom 201 itbygget
Til stede: Leonard Sandløkk, Alexander Holt
Frafall: Ingen
Møteleder: Leonard Sandløkk

Sak 01/23
Møteinnkalling godkjent.

Sak 02/23
Referat fra forrige møte gjennomgått og godkjent uten merknader.

Sak 03/23
Arbeidstimer er forventet å være 500 timer med en margin på 10%. Det betyr at forventet arbeidstid er minimum 450 timer til tross for utveksling, utsettelse og dårlig adgang til vizlab. Dersom kravet ikke er møtt betyr det ikke nødvendigvis at oppgaven blir underkjent men teller heller negativt på oppgaven og går utover helheten. Det anbefales å forsøke å få 450 arbeidstimer sammenlagt eller så mange som mulig.

Systemdokumentasjon og Kravdokumentasjon må ikke ha unødvendige eller irrellevante deler. Disse er maler som ikke må følges der det ikke gir mening. Spesifikke eksempler er klassediagram og sekvensdiagram. Bedre ting å ha med er oversikt over hvordan spillet funker slik at fremtidige brukere eller utviklere kan jobbe videre på oppgaven.

Sak 04/23
Statusrapportene ble slettet av en uheldig windowsoppdatering natten før men kopi av arbeidsloggen eksisterer og dermed skaper dette ikke mye problem. Statusrapporter tidligere besto også av fysiske møter med spilltesting.

Sak 05/23
Diskusjon om hva som er relevant å ha med I hovedrapporten og I hvilke deler forskjellige tema skulle være I.

Sak 06/23
Arbeid med skriving og hovedrapport frem til neste møte.

Sak 07/23 E
Neste møtet må flyttes tidligere til fredag 26 mai ettersom Alexander er I tyskland tirsdag 30 mai. Tidligere samme fredag ønskes det at en presentasjon holdes I forbindelse med bacheloroppgave for å vise frem prosjektet, hva som er gjort, og vise hva vizlabben brukes til. Mer detaljer kommer.

Trondheim 16.05.23 Leonard

Møtereferat 9

Referat fra prosjektmøte bacheloroppgave 90

dato/tid: 26.05.23 kl 13:45-15:45, Vizlabben
Til stede: Leonard Sandløkk, Alexander Holt
Frafall: Ingen
Møteleder: Leonard Sandløkk

Sak 01/23
Møteinnkalling godkjent.

Sak 02/23
Referat fra forrige møte gjennomgått og godkjent uten merknader.

Sak 03/23
Statusrapporter gjennomgått uten merknad

Sak 04/23
Viktige ting nevnt I hovedrapporten
Forskningsspørsmål
Plassering av segmenter
Teori relevans

Sak 06/23
Planlagt å ha et ferdig første utkast levert til veileder for sjekking Fredag 2 Juni 17:00.

Sak 07/23
Neste møte planlagt å være Mandag 5 Juni 2023.

Trondheim 25.05.23 Leonard

Møtereferat 10

Referat fra prosjektmøte bacheloroppgave 90

dato/tid: 08.06.23 kl 11:00-12:00, https://meet.idi.ntnu.no/leonard
Til stede: Leonard Sandløkk, Alexander Holt
Frafall: Ingen
Møteleder: Leonard Sandløkk
Sak 01/23
Møteinnkalling godkjent

Sak 02/23
Referat fra forrige møte gjennomgått og godkjent uten merknader.

Sak 03/23
Statusrapporter gjennomgått uten merknad

Sak 04/23
Hovedrapporten hadde god språk og argumenter. Anbefalt å endre struktur, legge til bilder og figurer, fikse på en kilde. Mye smådetaljer I forskjellige avsnitt for eksempel om VR teknologi historie. Snakket generellt om oppgaven og feilene som veileder fant og anbefalinger for hva som kan rettes og endres.

Sak 06/23
Finlesing av oppgaven, fikse det som er nevnt I sak 04. Gjøre spillet klar til å levere.

Sak 07/23
Siste møte før innleveringsfristen og er dermed siste møte.

Trondheim 25.05.23 Leonard

Timelister

Oppsummering av timelister

Ukenr	"Person nr 1"
2	0
3	0
4	1
5	0
6	0
7	0
8	0
9	1.5
10	7
11	24
12	10.5
13	11
14	31
15	13
16	45
17	3.5
18	16
19	48.5
20	60.5
21	65.5
22	67
23	80.5
Sum antall timer pr person/totalt	**485.5**

Uke 4

Timeliste		Leo
Aktivitet	**Kategori**	**Antall timer**
Møte	Møter	1.0
Ukesum uke 4		**1.0**

Uke 9

Timeliste		Leo
Aktivitet	**Kategori**	**Antall timer**
Møteinnkalling	Møter	0.5
Møtereferat	Prosesshåndbok	1.0
Ukesum uke 9		**1.5**

Uke 10

Timeliste		Leo
Aktivitet	**Kategori**	**Antall timer**
Møte	Møter	0.5
Arbeidslogg	Prosesshåndbok	0.5
Epost til Grethe	Eposter	0.5
Møtereferat	Prosesshåndbok	1.0
Prosjektplan	Forprosjektplan	4.0
Møteinnkalling	Prosesshåndbok	0.5
Ukesum uke 10		**7.0**

Uke 11

Timeliste		Leo
Aktivitet	**Kategori**	**Antall timer**
Møte	Møter	0.5
Møtereferat	Prosesshåndbok	1.0
Forprosjektplan	Forprosjektplan	9.0
Visjonsdokument	Visjonsdokument	9.0
Poster/Plakat	Post/Plakat	4.0
Møteinnkalling	Prosesshåndbok	0.5
Ukesum uke 11		**24.0**

Uke 12

Timeliste		Leo
Aktivitet	**Kategori**	**Antall timer**
Kravdokumentasjon	Kravdokumentasjon	5.50
Poster/plakat	Post/plakat	4.0
Møte	Møter	0.5
Møteinnkalling	Prosesshåndbok	0.5
Ukesum uke 12		**10.5**

Uke 13

Timeliste		Leo
Aktivitet	**Kategori**	**Antall timer**
Arbeid på spillet	Spillutvikling	10.0
Møte	Møter	0.5
Møteinnkalling	Prosesshåndbok	0.5
Ukesum uke 13		**11.0**

Uke 14

Timeliste		Leo
Aktivitet	**Kategori**	**Antall timer**
Arbeid på spillet	Spillutvikling	29.5
Kravdokumentasjon	Kravdokumentasjon	1.0
Møteinnkalling	Prosesshåndbok	0.5
Ukesum uke 14		**31.0**

Uke 15

Timeliste		**Leo**
Aktivitet	**Kategori**	**Antall timer**
Arbeid på spillet	Spillutvikling	11.0
Møte	Møter	1.0
Eposter	Eposter	0.5
Møteinnkalling	Prosesshåndbok	0.5
Ukesum uke 15		**13.0**

Uke 16

Timeliste		Leo
Aktivitet	**Kategori**	**Antall timer**
Arbeid på spillet	Spillutvikling	45.0
Ukesum uke 16		**45.0**

Uke 17

Timeliste		Leo
Aktivitet	**Kategori**	**Antall timer**
Arbeid på spillet	Spillutvikling	2.5
Eposter	Eposter	0.5
Møteinnkalling	Prosesshåndbok	0.5
Ukesum uke 17		**3.5**

Uke 18

Timeliste		Leo
Aktivitet	**Kategori**	**Antall timer**
Arbeid på spillet	Spillutvikling	14.0
Møte	Møter	1.0
Møteinnkalling	Prosesshåndbok	0.5
Forprosjektplan	Forprosjektplan	0.5
Ukesum uke 18		**16.0**

Uke 19

Timeliste		**"Person nr 1"**
Aktivitet	**Kategori**	**Antall timer**
Arbeidslogg	Prosesshåndbok	1.0
Møtereferat	Prosesshåndbok	2.0
Dokumentasjon av kode med doxygen	Spillutvikling	4.0
Informasjonsinnhenting/lesing av tidligere oppgaver	Informasjonsinnhenting	23.0
Visjonsdokument	Visjonsdokument	1.0
Hovedrapport	Hovedrapport	16.5
Eposter	Epost	1.0
Ukesum uke 19		**48.5**

Uke 20

Timeliste		Leo
Aktivitet	**Kategori**	**Antall timer**
Møte	Møter	1.0
Møtereferat	Prosesshåndbok	1.0
Arbeidslogg	Prosesshåndbok	4.5
Hovedrapport	Hovedrapport	33.0
Møteinnkalling	Prosesshåndbok	0.5
Kravdokumentasjon	Kravdokumentasjon	1.5
Forprosjektplan	Forprosjektplan	0.5
Informasjonsinnhenting	Informasjonsinnhenting	18.5
Ukesum uke 20		**60.5**

Uke 21

Timeliste		Leo
Aktivitet	**Kategori**	**Antall timer**
Spilltesting	Spillutvikling	1.0
Bilder	Ta bilder	1.5
Hovedrapport	Hovedrapport	55.5
Presentasjon	Presentasjon	5.0
Møte	Møter	1.0
Møtereferat	Prosjekthåndbok	1.0
Møteinnkalling	Prosjekthåndbok	0.5
Ukesum uke 21		**65.5**

Uke 22

Timeliste		Leo
Aktivitet	**Kategori**	**Antall timer**
Hovedrapport	Hovedrapport	67.0
Ukesum uke 22		**67.0**

Uke 23

Timeliste		"Person nr 1"
Aktivitet	**Kategori**	**Antall timer**
Hovedrapport	Hovedrapport	69.5
Møte	Møte	1.0
Møtereferat	Prosjekthåndbok	0.5
Møteinnkalling	Prosjekthåndbok	0.5
Sjekking av build/gjøre utviklingsdel innleveringsklar	Spillutvikling	4.0
Forprosjektplan	Forprosjektplan	1.0
Prosjekthåndbok	Prosjekthåndbok	1.0
Visjonsdokument	Visjonsdokument	1.0
Kravdokumentasjon	Kravdokumentasjon	1.0
Refleksjonsnotat	Refleksjonsnotat	1.0
Ukesum uke 23		**80.5**

Vision Document

Pac-Man in VR on the Omnideck
Visjonsdokument

Versjon <1.2>

Revisjonshistorie

Dato	Versjon	Beskrivelse	Forfatter
16/03/2023	1.0	Førsteutkast	Leonard Sandløkk Schiller
10/5/2023	1.1	Andreutkast	Leonard Sandløkk Schiller
11/06/2023	1.2	Finjustering før levering	Leonard Sandløkk Schiller

Innholdsfortegnelse

Innledning 2
Sammendrag problem og produkt 2
Problemsammendrag 2
Produktsammendrag 2
Overordnet beskrivelse av interessenter og brukere 3
Oppsummering interessenter 3
Oppsummering brukere 3
Brukermiljøet 3
Sammendrag av brukernes behov 3
Alternativer til vårt produkt 4
Produktoversikt 4
Produktets rolle i brukermiljøet 4
Forutsetninger og avhengigheter 4
Produktets funksjonelle egenskaper 4
Ikke-funksjonelle egenskaper og andre krav 4

Innledning

Denne oppgaven utføres av Leonard Sandløkk Schiller som bachelorprosjekt i ingeniørfag innenfor data ved NTNU Trondheim. Prosjektet er basert på oppgaven gitt av Alexander Holt ved NTNU om "PacMan og/eller Frogger i VR". Formålet med utviklingsdelen av oppgaven er å lage et VR-spill som bruker Omnidecken for bevegelse i spillet. Denne tredemølla er i VR Labben på Gløshaugen. I rapporten ønsker jeg å lage retningslinjer og forslag til videre spillutvikling i et lignende VR miljø med Omnideck som bevegelseskontroller. Disse vurderingene vil komme gjennom arbeidet og funnene gjort i løpet av arbeidsprosessen. I tillegg ønsker jeg å drøfte hvordan å overføre 2d spill til VR og dokumentere hvordan jeg gjorde dette med denne utviklingsoppgaven.
Jeg valgte denne oppgaven fordi jeg har interesse og tidligere erfaring med både VR og spillutvikling.

Sammendrag problem og produkt

Problemsammendrag

Problem med	Et evigvarende problem med VR er hvordan man skal skape en immersiv VR opplevelse samtidig som man gir en god spillopplevelse.
berører	Dette påvirker alle som spiller i VR.
som resultatet av dette	Dette fører til en verre spillopplevelse og en lavere grad av immersjon.
en vellykket løsning vil	Øke spillerens immersjon i spillet og gi brukeren en bedre spillopplevelse.

Produktsammendrag

For	Produkteier
som	Har behov for et demo-spill for Omnideck
produktet navngitt	VR PacMan
som	Er en ny versjon av et klassisk spill, i et VR miljø med nye teknologier
I motsetning til	Konvensjonelle spill i VR
Har vårt produkt	Omnideck for bevegelse i VR rom

Overordnet beskrivelse av interessenter og brukere

Oppsummering interessenter

Navn	Utdypende beskrivelse	Rolle under utviklingen
Alexander Holt	Representerer oppdragsgiver og produkteier som spør om	Oppdragsgiver, Veileder

	produktet og setter krav til oppgaven.	
Leonard Sandløkk Schiller	Utvikler og oppgavetaker. Har ansvar for å utvikle og ferdigstille produktet etter kravene.	Utvikler, Oppgavetaker

Oppsummering brukere

Navn	Utdypende beskrivelse	Rolle under utviklingen	Representert av
Sluttspiller	Sluttbrukeren eller sluttspilleren er de som spiller spillet når den er ferdigutviklet.	Frivillige spillere vil prøve et mvp. under utviklingen og gi forslag til forbedringer i spillet.	Seg selv
Utvikler	Utvikler er nødvendig for initial og kontinuerlig utvikling av spillet.	Utvikleren vil utvikle spillet.	Seg selv

Brukermiljøet

Brukermiljøet I dette tilfellet er å spille VR spill. Produktet vil være enda et spillvalg for brukeren å velge mellom, men vil skille seg ut fordi den er ment til å spilles med bruk av Omnideck. Dette medfører en ekstra arbeidsprosess I form av å gå på Omnideck mens brukeren spiller.

Sammendrag av brukernes behov

Behov	Prioritet	Påvikrer	Dagens løsning	Foreslått løsning
VR spill med mer immersjon og bedre kontroller løsning.	1	Spilleren	Konvensjonelle VR spill bruker ikke tredemølle og har lavere immersjon.	Utvikle VR spill som bruker tredemølle.

Alternativer til vårt produkt

Sluttproduktet er et spill og derfor er alternativt produkt alle andre eksisterende spill. Dette gjelder både spill som er og ikke er i VR. Samtidig er spill i VR begrenset, og det finnes spesielt få eller ingen spill som bruker tredemølle i samsvar med VR.

Produktoversikt

Produktets rolle i brukermiljøet

Figur 1: Bruker går på Omnideck

Forutsetninger og avhengigheter

Noen av forutsetningene beskrevet i forprosjektsplanen inkluderer tilgang til utviklingsprogram, i dette tilfellet Unity. I tillegg er det forutsatt tilgang til VRLabben på NTNU med tredemølla slik at det er mulig å bruke den for testing av spillet.

Produktets funksjonelle egenskaper

Spillets krav:
Må kunne spilles i VR med VR briller
Må kunne spilles med tredemølle i VR
Full liste over krav for spillet star I kravdokumentasjonen (Kravdokument)

Ikke-funksjonelle egenskaper og andre krav

FURPS:
Functionality
Spillet skal følge alle kravene stillt under funksjonelle egenskaper og skal kunne spilles av minst en spiller om gangen. Spillet skal kunne spilles så mange ganger brukeren ønsker.
Usability
Spillet skal ha en forståelig og enkel interface som er intuitiv for brukeren. Det er essensiellt at spillet er konsistent og responsiv ettersom det er et spill i VR. Uten de to attributtene vil spillets kvalitet drastisk synke og det kan skape større problemer mht. VR sykdom.
Endra start knapp så det var enklere å trykke
Valgte 3 hastigheter for simplisitet

Reliability
Spillet skal være nedlastbar. Spillet skal ikke kræsje mye ettersom det senker spillbarheten og spillopplevelsen.
Spillet skal ikke ha bugs, glitches eller exploits, dersom noen er funnet må de fikses.

Performance
Det er essensiellt at spillet har en relativ høy skjermhastighet og oppdateringsfrekvens for spillbarhet, spillopplevelse, kvalitet og for å redusere vr sykdom. Det er relevant å ha noe fokus på spillets optimalisering slik at den ikke krever for mange ressurser.
Supportability
Spillet skal prøve å gjøres enkel å vedlikeholde med god dokumentasjon.

Requirements Document

Pac-Man in VR on the Omnideck
Kravdokumentasjon

Versjon <1.3>

Revisjonshistorie

Dato	Versjon	Beskrivelse	Forfatter
<21/03/23>	<1.0>	<Førsteutkast>	<Leonard>
<04.04.23>	<1.1>	<Spøkelser>	<Leonard>
<15.04.23>	<1.2>	<Gjenstander>	<Leonard>
<18.05.23>	<1.3>	<Spillforklaring>	<Leonard>

Innholdsfortegnelse

Introduksjon 1
User Stories 2
Spilloversikt 5
Prototyper 7
Referanser..9

Introduksjon

Dette dokumentet inneholder kravdokumentasjon for spillet som lages i samsvar med bacheloroppgave 90, VR PacMan. Denne dokumentasjonen er nødvendig for å dokumentere alle ønsket krav for spillet og virker som en ressurs for å kunne etterfølge og bekrefte at alle ønsket tekniske krav I spillet er oppfylt. Dokumentet inneholder "User Stories" som beskriver brukerens ønsker, behov og mål i spillet. Disse skal ha akseptansekriterier som, hvis oppnådd, tilsier et oppfylt teknisk krav.
Dette dokumentet viser en oversikt av hva som skal skje i spillet.
Dette dokumentet inneholder IKKE domenemodeller eller sekvensdiagrammer.
Disse er ofte brukt I kravdokumentasjon, spesielt for nett-baserte tjenester eller konvensjonelle applikasjoner og bidrar til bedre kommunikasjon I løpet av utviklingsprosessen og en god strukturell oversikt av systemet.
I denne oppgaven er de ikke gunstige å ha med for flere grunner.
Gruppa består av kun et medlem, og dermed er det umulig med kommunikasjonsfeil internt i gruppa både i kreative prosessen og i utviklingsfasen.

Ettersom at PacMan er et eksisterende, godt kjent, og relativt enkelt spill, er sannsynligheten for feilkommunikasjon mellom utviklingsgruppa og oppdragsgiver lav. Dermed er nødvendigheten for domenemodell minket.
Utviklingsoppgaven består av å lage spillet PacMan. Spillutviklere bruker ofte entitet-komponent arkitektur istedenfor domene arkitektur. I denne oppgaven ble unity brukt som også bruker entitet-komponent arkitektur. (Unity, n.d.)
Dette begrenser hvilke modeller som er mulig å lage for applikasjonen, ettersom flere modeller som kan lages for domene drevet design kan ikke lages eller er ikke nyttige å lage for entitet-komponent drevet design. Dette kommer fra arkitekturforskjellen hvor entitet-komponent arkitekturen ikke er hierarkisk, men heller komponent basert. (Nystrom, 2014)

Til slutt vil utviklingsprosessen ligne mer på FDD (feature driven development) i motsetning til DDD (domain driven design). FDD setter mer fokus på produksjon av egenskaper fremfor å modellere domenet.

User Stories

Overordnet bruker:
Spilleren
Overordnet mål:
Spille PacMan i VR med Omnideck

Som spilleren
Ønsker jeg å spille hele spillet i VR
Fordi dette er en essensiell spillegenskap jeg leter etter i spillet

- Spillet må kunne spilles i sin helhet i VR
- Spillet skal spilles med VR briller og VR kontrollere (for eksempel HTC VIVE)

Som spilleren
Ønsker jeg å være i en labyrint som ligner den i original PacMan spillet
Fordi dette gjør spillet mer som originale PacMan

- Labyrinten må ha gulv og vegger, disse skal være blå og svart som originale PacMan veggene og "gulvet"
- Veggene skal ha rimelige proporsjoner og ikke se forvrengt ut
- Det skal ikke være mulig å gå gjennom veggene fordi dette er ikke mulig i original spillet
- Veggene skal være høyere enn spilleren, slik at man ikke kan se over veggen
- Veggene og gulvet skal stoppe andre entiteter, og ikke la dem passere gjennom
- Veggene og gulvet skal ikke kunne bevege på seg
- I hver horisontal ende av labyrinten skal det være mulig å teleportere til andre siden slik som i original spillet

Som spilleren
Ønsker jeg å kunne bevege meg i spillet ved å bevege meg på Omnideck tredemøllen
Fordi dette er en essensiell spillegenskap jeg leter etter i spillet

- All bevegelse i spillet må skje gjennom Omnideck tredemølla
- Når jeg beveger meg på tredemølla, skal karakteren min bevege meg i spillet
- Bevegelsen min i spillet skal ha samme retning og fart som bevegelsen min på tredemølla
- Hastigheten både på tredemølla og i spillet må være rimelig for meg som spiller

Som spilleren
Ønsker jeg å samle pellets og power pellets
Fordi dette er en essensiell del av det originale PacMan spillet

- Når jeg går inn i en pellet skal:
  - Pelleten forsvinne
  - Jeg få 10 poeng
  - En lyd spilles av
- Når jeg går i en power pellet skal:
  - Power pelleten forsvinne
  - Jeg få 50 poeng
  - Spøkelser bli skremt
  - En lyd spilles av
  -

Som spilleren
Ønsker jeg at det er spøkelser i labyrinten
Fordi dette er en essensiell del av det originale PacMan spillet

- Spøkelsene må ha forskjellige farger, rød, cyan, rosa, oransj
- De må ha samme modell og ligne på hverandre utenom farge

Som spilleren
Ønsker jeg å bli jaget av spøkelser
Fordi dette er en essensiell del av det originale PacMan spillet

- Spøkelsene må jage etter meg
  - Hvert spøkelse skal ha en oppførsel slik som i originale PacMan spillet, denne oppførselen består av to faser, jaging og flykting
  - I Jage fasen skal:
    - Blinky (rød) gå direkte til spilleren
    - Pinky(rosa) gå foran der spilleren går
    - Inky(blå) gå til dit Blinky ville vært speilet om spilleren
    - Clyde(oransj) gå til dit spilleren er, frem til han er innenfor en hvis omkrets og deretter gå til punktet han går i flykte fasen
  - I flykte fasen skal:
    - Blinky(rød) gå til øvre høyre hjørne
    - Pinky(rosa) gå til øvre venstre hjørne
    - Inky(blå) gå til nedre høyre hjørne
    - Clyde(oransj) gå til nedre venstre hjørne
- De navigere gjennom labyrinten og finne veien til meg
- De skal ha en rimelig fart slik at det er mulig å løpe vekk fra dem, men uten at de er for sakte
- De skal rotere slik at de alltid ser fremover i retningen de beveger seg i

Som spilleren
Ønsker jeg å kunne skremme spøkelsene
Fordi dette er en essensiell del av det originale PacMan spillet

- Jeg må kunne skremme spøkelsene for eksempel ved å spise en power pellet
- Spøkelsene må alle endre farge til mørkeblå og øynene og munn må endres slik at det vises tydelig at de er redde
- Når spøkelsene blir redde må de endre oppførsel, de må:
  - Slutte å jage meg
  - Begynne å navigere vekk fra meg

Som spilleren

Ønsker jeg å kunne fange skremte spøkelser
Fordi dette er en essensiell del av det originale PacMan spillet

- Dersom spøkelset er skremt skal jeg kunne bevege meg inn i spøkelset, da skal:
  - En lyd spilles av
  - Spøkelses "kroppen" bli usynlig
  - Øynene skal forbli synlige
  - Jeg skal få poeng basert på hvor mange spøkelser jeg har tatt
  - Spøkelset skal øke hastighet
  - Spøkelset skal navigere til midten av labyrinten
  - Når spøkelset kommer til midten av labyrinten skal den "gjenopplives" og bli til et vanlig spøkelse igjen, og gjenoppta vanlige spøkelses aktiviteter

Som spilleren
Ønsker jeg å kunne se min totale oppsamlet poengsum
Fordi dette er en essensiell del av det originale PacMan spillet

- Poengsummen skal vise en sanntidsoppdatert total poengsum av alle poeng samlet i løpet av spillet
- Denne poengsummen skal bli vist på en måte som kun er mulig i VR for å styrke VR opplevelsen (i motsetning til flat på skjermen som i original spillet)

Som spilleren
Ønsker jeg å ha et kart jeg kan se på
Fordi da vil jeg ha en ny og bedre måte å navigere meg i labyrinten

- Kartet skal oppdateres i sanntid slik at alle bevegelser og ting som skjer i spillet også skjer på kartet
- Kartet skal vise hele labyrinten ovenfra og vise alle entiteter i labyrinten
- Kartet skal vises på hånda til spilleren
- Kartet skal kunne skrus av og på av spilleren
- Kartet skal ha en rimelig størrelse for å ikke blokkere spillerens syn, men likevel stor nok til å vise detaljer

Som spilleren
Ønsker jeg å ha en meny
Fordi jeg vil endre spillets innstillinger

- Jeg må kunne åpne og lukke menyen når jeg vil
- Menyen må ha forskjellige innstillinger som:
  - Endre hastigheten min
  - Endre hastighet til spøkelsene
  - Skru kartet av og på
  - Skru lyd av og på
- Når jeg åpner menyen skal spillet stå i pause slik at ingenting skjer i spillet
- Menyen skal ha valg om å avslutte eller fortsette spillet
- Det skal være mulig å åpne resultattavla

Som spilleren
Ønsker jeg å ha en resultattavle
Fordi jeg vil se min egen poengsum og kunne sammenligne med andre poengsummer

- Jeg må kunne åpne og lukke resultattavla gjennom menyen
- Resultattavla må åpnes når spillet avsluttes
- Resultattavla må ha flere resultater og de skal være de høyeste oppnådde resultatene

Som spilleren
Ønsker jeg å kunne samle frukt på banen
Fordi dette er en del av de originale PacMan spillet

- Fruktene skal instansieres over tid
- Det skal kunne være en frukt på banen om gangen
- Når jeg går inn i frukten skal:
  - Jeg få poeng basert på frukten
  - En lyd spilles av
  - Frukten droppe en gjenstand som jeg kan plukke opp og bruke på forskjellige måter

Som spilleren
Ønsker jeg å se hvor mange liv jeg har
Fordi dette er en essensiell del av det originale PacMan spillet

- Livene skal være sanntidsoppdatert
- Disse livene skal bli vist på en måte som kun er mulig i VR for å styrke VR opplevelsen (i motsetning til flat på skjermen som i original spillet)

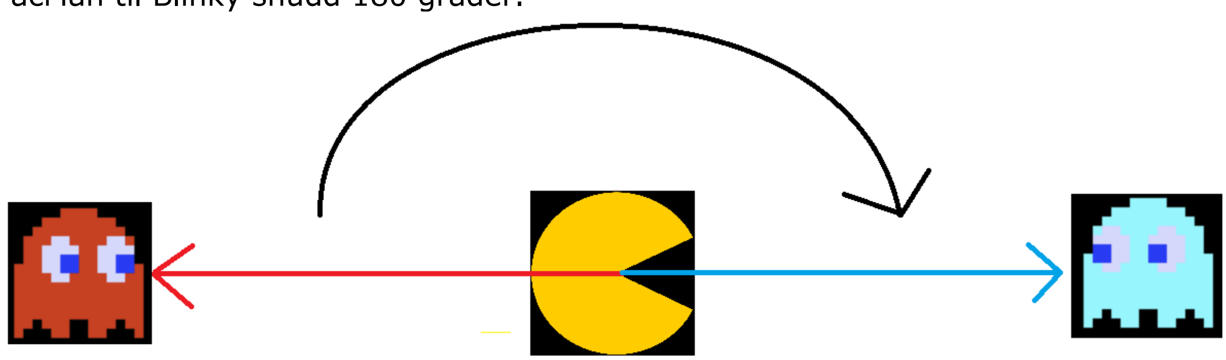
Som spilleren
Ønsker jeg at vanskelighetsgraden øker over tid og med antall nivåer jeg fullfører
Fordi dette skjer i det originale PacMan spillet

Spilloversikt

Før spillet
Når spilleren åpner spillet, skal de først havne i en "lobby" hvor spillet ikke enda skal starte. "Lobbyen" skal være som et pauserom der spilleren kan velge innstillinger, gå rundt på egenhånd, og få en følelse for kontrollene før spillet starter. Når spilleren er klar så skal han kunne trykke en knapp for å starte spillet.

Spillstart
Når spilleren starter spillet skal de overføres til spill senen. Umiddelbart er det flere prosesser som starter.
Pellets
I det spillet begynner vil banen bli fullt opp av vanlige Pellets og Power Pellets. Dersom spilleren går inn i en vanlig pellet økes poengsummen med 10 poeng og en lyd vil spilles av. Dersom spilleren går inn i en Power Pellet vil dette øke poengsummen med 50 poeng, en lyd vil spilles av og skremme alle spøkelser.
Gjenstander
Når spillet starter, vil gjenstander begynne å plasseres tilfeldig i labyrinten. Kun en gjenstand blir lagt ut om gangen. Disse gjenstandene har forskjellige effekter når de brukes av spilleren. For eksempel kan en gjenstand brukes til å skremme, spise, endre hastigheten til, eller blokkere spøkelser. For å bruke gjenstanden må spilleren gå inn i gjenstanden, da droppes en ny gjenstand som spilleren kan plukke opp. Gjenstandene brukes på forskjellige måter, men ofte kan de bli kastet i bakken eller brukt som pistol for å brukes.
Spøkelser
Ved spillstart skal fire spøkelser av de forskjellige fargene lages på forskjellige steder på kartet. Spøkelser bruker en navmesh for å bevege seg til et bestemt

punkt. Dette punktet blir endret mange ganger for å endre hvor spøkelset beveger seg og oppførsel.
Disse spøkelsene har 4 forskjellige moduser. Disse er: chase, scatter, scared, eaten.
Til å begynne med starter alle spøkelser i scatter modus. Scatter og chase modus er ganske like. I disse modusene er spøkelsene farget og går til sine spesifikke punkter i labyrinten. I chase modus vil hvert spøkelse gå mot et punkt som generelt er nært spilleren og i scatter modus vil hvert spøkelse gå mot sitt eget flyktnings punkt i hvert hjørne. Disse to modusene blir endret med tid basert på en intern modus klokke for å bedre etterligne det originale PacMan spillet og for å variere spøkelsenes oppførsel. Denne klokka må ha lignende tider som den originale PacMan modus klokka har.
Dersom spilleren går inn i en Power Pellet eller bruker en gjenstand som skremmer spøkelser vil alle spøkelser som ikke er i scared eller eaten modusen gå over til scared modusen. I scared modusen blir spøkelset mørkeblå, øynene endrer fasong, og det finner et tilfeldig punkt i labyrinten som er vekk fra spilleren og går til det punktet, nye punkter blir funnet når de er fremme. I denne modusen kan spøkelset spises ved at spilleren går inn i dem. Når dette skjer, går spøkelset over i eaten modus, spilleren får poeng basert på hvor mange spøkelser ble spist, og en lyd vil spilles av.
I eaten modus vil kroppen til spøkelset bli usynlig, og den går umiddelbart til midten av labyrinten. Når den er fremme, vil den bli "gjenopplivet" og gå over til enten chase eller flykt modus basert på interne modus klokka.
Hvert spøkelse har eget navn, oppførsel, og punkter i labyrinten. Det røde spøkelset heter Blinky og er en jager. Når han er i chase modus er punktet han går til der spilleren står, og han vil gå direkte til spilleren. I scatter modus vil han bevege seg til et punkt på øverste høyre del av labyrinten. Det rosa spøkelset heter Pinky og er en bakholdsangriper. Når hun er i chase modues er punktet hun går til foran der spilleren står, og hun vil gå direkte foran spillerens posisjon. I scatter modus vil hun bevege seg til et punkt på øverste venstre del av labyrinten. Det blå spøkelset heter Inky og er en lagspiller. Når han er i chase modus er punktet han går til basert på Blinky dersom Blinky er i chase eller scatter modus, og direkte der spilleren står ellers. Når Blinky er i chase eller scatter modus vil Inky sitt punkt være vektoren fra PacMan til Blinky snudd 180 grader.



Dette betyr også at når Blinky nørmer seg spilleren vil Inky og.

Siden Inky beveger seg mot PacMan fra motsatt retning av Blinky så funker det som et tangangrep på spilleren og dette samspillet er årsaken til at Inky er en lagspiller. I scatter modus vil han bevege seg til et punkt på nederste venstre del av labyrinten.
Det siste spøkelset er oransj og heter Clyde. Han er lat. Når han er i chase modus er punktet han går til der spilleren står, og han vil gå direkte til spilleren, men når han nærmer seg spilleren vil han bevege seg til samme punkt som i scatter modus. I scatter modus vil han bevege seg til et punkt på nederste venstre del av labyrinten.
Alle av spøkelsene vil gå direkte til spilleren dersom de kan se spilleren.

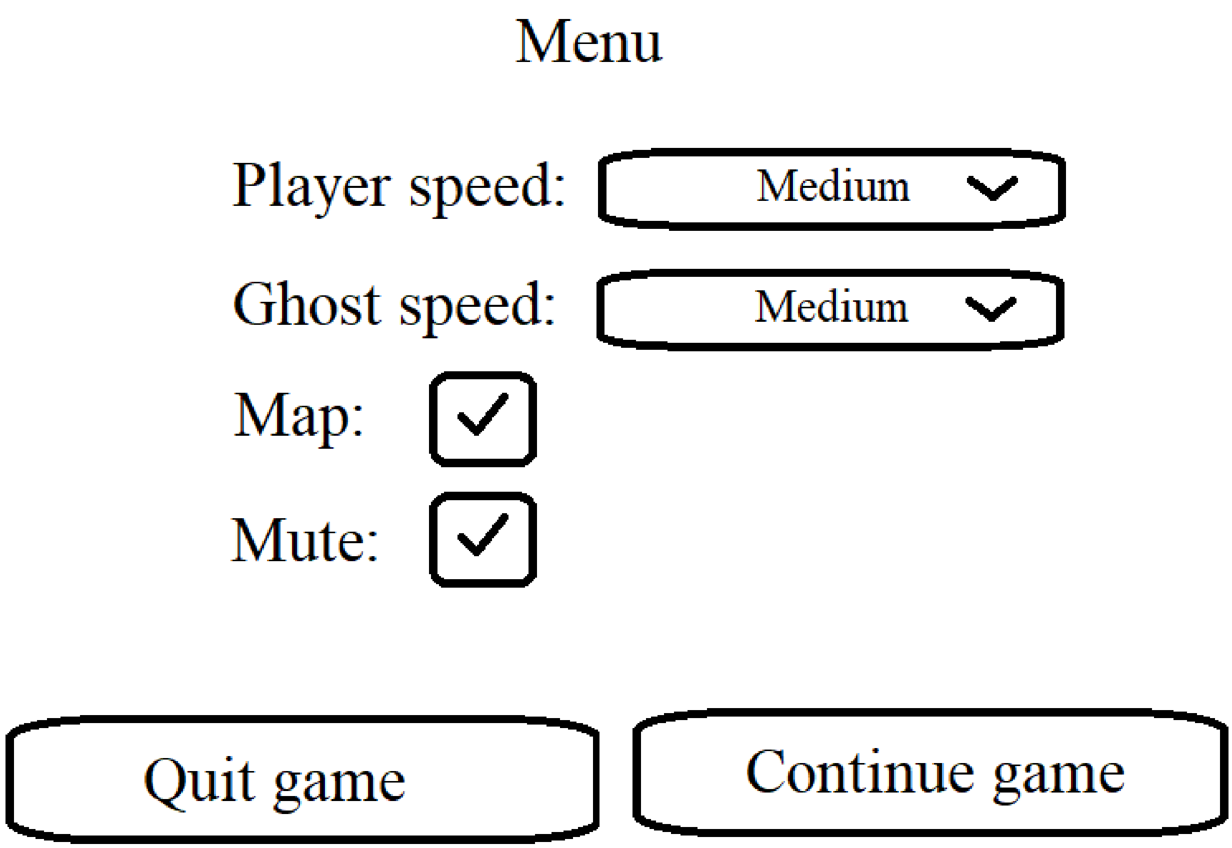
Måten spøkelser funker på ble funnet av videoen Pac-Man Ghost AI Explained av Retro Game Mechanics Explained. (Retro Game Mechanics Explained, 2019)

Ved å trykke på midterste knapp skal spilleren kunne åpne hovedmenyen. Dette skal stanse spillets klokker, stanse bevegelse av både spiller og spøkelser, og ellers stanse ting som skjer i spillet. Når spilleren trykker på meny knappen skal menyen vises på skjermen. Spillet skal starte igjen når spilleren lukker menyen.

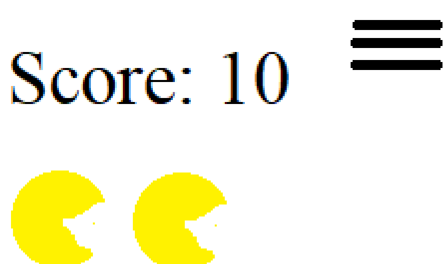
Prototyper

Wireframes

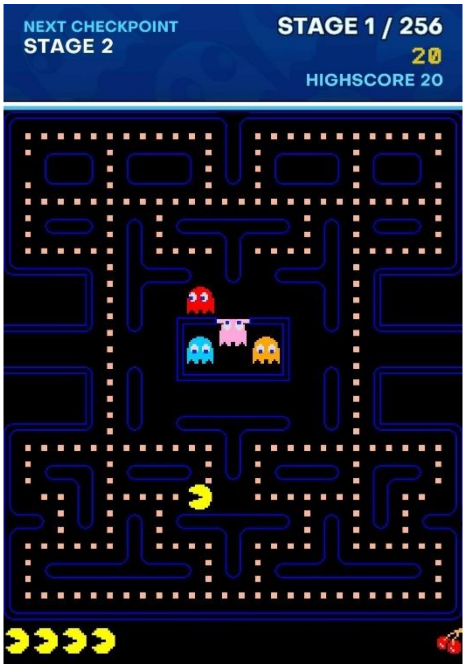
Meny



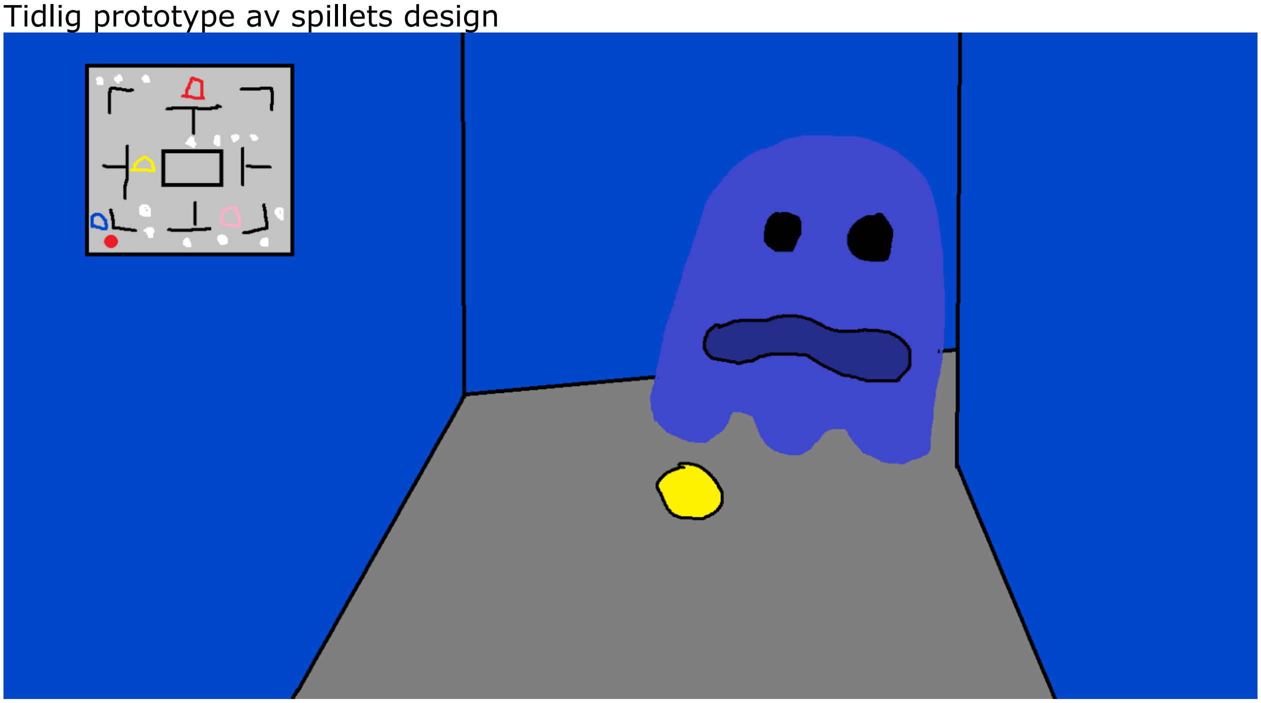
Poengsum og liv (venstre arm)



Kart (Høyre hånd)



Tidlig prototype av spillets design



Referanser

Nystrom, R. (2014). Component. In R. Nystrom, *Game Programming Patterns* (p. 354). Genever Benning. Retrieved April 2023, from https://gameprogrammingpatterns.com/

Unity. (n.d.). *Entity Component System*. Retrieved April 2023, from Unity Manual: https://docs.unity3d.com/Packages/com.unity.entities@0.1/manual/index.html

Retro Game Mechanics Explained. (2019, July 13). Pac-Man Ghost AI Explained. Retrieved March 2023, from https://www.youtube.com/watch?v=ataGotQ7ir8

System Document

Necessary software:

unityhub 3.4.1

https://unity.com/download

unity version 2021.3.16f1 (LTS)

(Downloaded in unityhub)

vive wireless app 1.20191127.1

https://www.vive.com/hk/support/wireless-adapter/category_howto/about-vive-wireless.html

steamvr 1.25.8

https://store.steampowered.com/app/250820/SteamVR/

omnitrack app v2.5.12-Beta

proprietary software, comes with the Omnideck

How to install:

#1: Installing UnityHub and correct unity version

Install Unityhub version 3.4.1 from https://unity.com/download

Click download for windows, this will download UnityHubSetup

Run UnityHubSetup, agree to TOS, click install, finish and run UnityHub

Sign in to your unity account

If prompted to install newer version, click skip installation

Click install editor button

Select archive tab

click on download archive

On the website, click download LTS releases

Select 2021.3 and find version 2021.3.16f1

Click Unity Editor Windows (X86-64)

Close UnityHub

Open Unity 2021.3.16f1 Setup

Accept License agreement

Download

Open Unity 2021.3.16f1

UnityHub should open, verify that version 2021.3.16f1 is installed by clicking installs tab and verifying this version is there.

#2 Installing Steam and SteamVR

First install steam from https://store.steampowered.com/about/

This will install SteamSetup

Open SteamSetup, select language, install, finish and run Steam

Sign in to Steam Account

Search for SteamVR
https://store.steampowered.com/app/250820/SteamVR/

Download and open SteamVR

#3 Installing HTV Vive

Install vive wireless app from https://www.vive.com/hk/support/wireless-adapter/category_howto/about-vive-wireless.html

this will download vive wireless setup

open vive wireless setup

#4 Installing Omnifinity Omnitrack application

This is proprietary software that comes with the Omnideck

Follow installation guide from the installation manual

#5 Downloading gitlab repo

Open https://gitlab.stud.idi.ntnu.no/leonarss/leoomnibach/-/tree/main

Click clone button

Copy HTTPS link

Open command prompt

Navigate to preferred installation directory

type in git clone https://gitlab.stud.idi.ntnu.no/leonarss/leoomnibach.git

Opening application:

Method 1 with executable:

Open VRPacMan.exe

Open Vive wireless application, SteamVR and Omnitrack applications

Game should start and be playable!

-NOTE

The HMD might disconnect with this method for unknown reasons,

When turning on controllers pairing again helps with stability

Game DOES work in this mode though, simply HMD/controller disconnects

Method 2 in unity editor:

#1 Opening project in UnityHub

Open UnityHub

Select Open

Navigate to installed git repo, double click leoomnibach, then double click VRPacMan, then open

VRPacMan should appear as a project under projects tab

Select editor version 2021.3.16f1 within the project

Click on project to open

Do not install new version of unity editor if prompted

Click accept all for recommended project settings for steamvr

#2 Starting the game

Make sure Vive wireless application, SteamVR and Omnitrack applications are turned on and connected

Click play button in unity editor

Game should start and be playable!

How to play!

You spawn in the center of the maze in the lobby,

Here you can walk around to get used to the Omnideck and when you're ready you can press the start game button on your left hand with the laser on your right hand.

You will then spawn in a new maze with pellets, ghosts and items!

On your left hand you will see your score and lives, on your right, the map.

There are only three buttons,

TOUCHPAD -opens the menu

TRIGGER -can be used to grab items and shoot guns

SIDEGRIP -can be used to grab items and guns

There are several items in this game that behave differently to regular Pac-Man

The items are:

Cherry, Strawberry, Orange, Apple, Melon, Galaxian, Bell, and Key

When the player walks into these items they spawn new usable items

The Cherry is worth 100 points and the Galaxian is worth 200 points. Both spawn guns that can shoot at the ghosts

The Strawberry is worth 300 points and spawns a jar of jam that, when thrown, creates a jam spill and slows down ghosts

The Orange is worth 500 points and spawns an orange slice which can be used as a wall to block ghosts

The Apple is worth 700 and spawns an apple trap which stops a ghost right in it's tracks

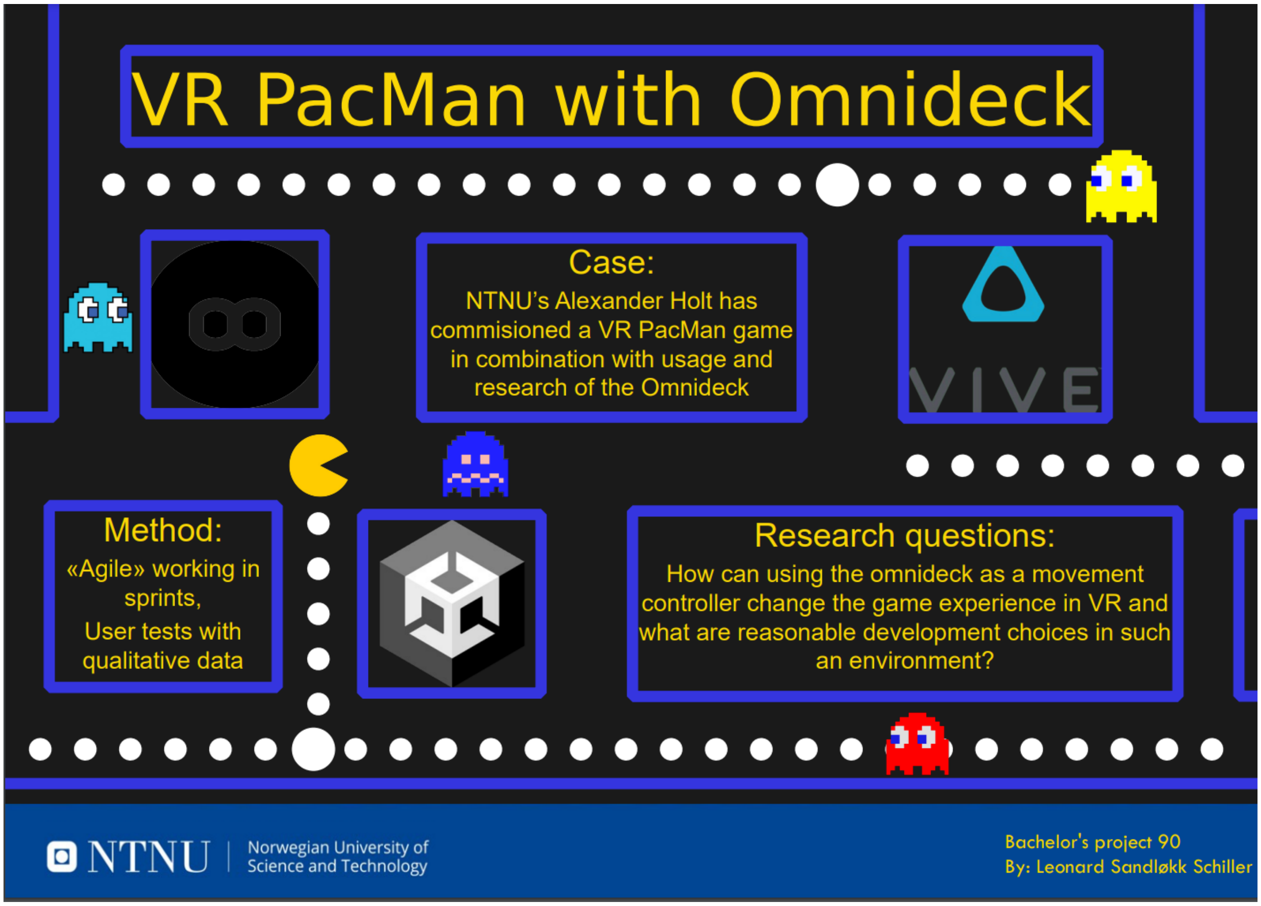
The Melon is worth 1000 and spawns a melon which can be thrown at ghosts to scare them

The Bell is worth 3000 and spawns a bell that scares all ghosts when thrown

The Key is worth 5000 points, gives the player an extra life, and spawns a key which can be thrown at ghosts to scare them

This game works like classic Pac-Man, have fun!

Poster Presentation



Denne ble presentert tidlig i oppgaven og forskningspørsmålene endret seg underveis.

Gitlab

https://gitlab.stud.idi.ntnu.no/leonarss/leoomnibach/-/tree/main

NTNU

Norwegian University of
Science and Technology