Post-mortem
Navigating design challenges for a VR game

MechVR

Riku Erkkilä &
Andrei Duván Rodríguez Baquero
MechVR Post-mortem
MechVR Post-mortem
Navigating design challenges for a VR game

Riku Erkkilä &
Andrei Duván Rodríguez Baquero

Master’s thesis prepared for the degree of Master of Arts in New Media: Game Design and Production

Department of Media
School of Arts, Design and Architecture
Aalto University

October 2019
Contents

INTRODUCTION 01
  1.1 VR and the game industry 03
  1.2 Short description of MechVR 04
  1.3 Division of work 05

THEORETICAL BACKGROUND 07
  2.1 About VR 09
    2.1 Game definitions 10
    2.2 Game genres 11
      2.2.1 MechVR as a first-person shooter game 11
      2.2.2 MechVR as a vehicle simulation game 13
    2.3 MDA game analysis framework 14
      2.3.1 Description of MDA framework 14
      2.3.2 MechVR viewed through the lens of the MDA framework 14

POST MORTEM 17
  3.1 Goals for the game 19
  3.2 Project timeline 19
  3.3 Main game design challenges 20
    3.3.1 Point of view: Pilot vs. Mech 20
    3.3.2 Reducing VR-induced sickness 22
    3.3.3 Designing the mech controls 24
      3.3.4 Finding the challenge 30
      3.3.5 Tinkering with the mech 32
  3.4 Mech design 34
    3.4.1 General design based on function 34
    3.4.2 Evolution of the design 35
    3.4.3 Multi-system character design 50
  3.5 Applying ergonomics in virtual reality games 51
    3.5.1 Human-centered VR design (Industrial design in VR) 51
    3.5.2 Designing the cabin and controls 52

REFLECTION 57
  4.1 Lessons learned 59
  4.2 Future work 60

CONCLUSION 63
Acknowledgements

I want to thank my mother Luz Marina Baquero and my family who have encouraged me throughout my life and supported all my crazy plans and ideas of going to the opposite side of the world to make games. Also my friend and thesis partner Riku for all the support and fun we had making this project and certainly projects to come. Thank you to Miikka Junnila for being our advisor and teacher all these years. Big thanks to all my “noob” friends from media lab and specially Akbar, Ilari and Lisa for helping us with proofreading and organizing this book, you guys are awesome!

Quiero dedicar este logro a mi abuela Concepción Hernández quien dejó este mundo antes de poder compartir este triunfo con ella, quiero agradecer a mi familia y especialmente a mi madre Luz Marina Baquero por todo el apoyo y soporte que me ha dado, nada habría sido posible sin ti, Gracias madre mía!

– Andrei Duván Rodríguez Baquero, 2019

I want to thank all the great people at Media Lab for the collaboration, help and inspiration and also my family, who have been tirelessly egging me on to finish a thesis or another for the last fifteen years.

– Riku Erkkilä, 2019
Abstract of master’s thesis

Kirjoittajat  Riku Erkkilä ja Andrei Duván Rodríguez Baquero
Lopputyön nimi  MechVR Post-mortem - Navigating design challenges for a VR game
Tutkinto  Taiteiden maisteri
Pääaine  Uuden Median Maisteriohjelma: Pelisuunnittelu ja -tuotanto

Vuosi  2019  Sivumäärä 86  Kielu  Englanti

Abstract

“MechVR” on tämän tutkielman kirjoittajien kehittämä prototyyppivideopeli, joka mukauttaa ensimmäisen persoonan ammuntapeligenren virtuaalitodellisuuteen.

Pelin ydinidea on laittaa pelaajahahmo ohjaamaan “mechiä”, eli suurta, ihmisenmuotoista panssarajoneuvoa samalla kuitenkin pyrkien säilyttämään ensimmäisen persoonan ammuntapeille tyypillinen pelattavuus. Tämä tutkielma dokumentoi pelin kehitysprosessin keskityen erityisesti suunnittelushaasteisiin, jotka liittyvät genrelle tyypillisten pelimekaniikkojen siirtämiseen virtuaalitodellisuuteen. Tutkielma kuvailee myös näistä haasteista seuranneita ajatusprosesseja ja suunnitteluratkaisuja.

Keywords  pelit, pelisuunnittelu, virtuaalitodellisuus, mech, ammuntapeli
Abstract of master’s thesis

Authors  Riku Erkkilä and Andrei Duván Rodríguez Baquero
Title of thesis  MechVR Post-mortem - Navigating design challenges for a VR game
Degree  Master of Arts
Degree programme  MA in New Media: Game Design and Production

Year of approval  2019  Number of pages  86  Language  English

Abstract
“MechVR” is a prototype game developed by the authors that adapts the first-person-shooter game genre for virtual reality.

The key idea of the game is to place the player's character inside a “mech”, a large, bipedal, anthropomorphic armored vehicle while still retaining the gameplay typical for a first-person-shooter game. This thesis documents the game development process, with a focus on the design challenges of adapting the game mechanics typical for the genre to a virtual reality environment. It also outlines the thought processes and design choices that resulted from these challenges.

Keywords  games, game design, virtual reality, mech, first-person-shooter
List of figures

Figure 1.1: MechVR’s protagonist’s mech seen from the front
Figure 1.2: Player’s mech facing two enemies
Figure 2.1: id Software. (1992). Wolfenstein 3D
Retrieved from: https://twitter.com/3DRealms/status/992817007686021120
Figure 2.2: Far Cry 5 by Ubisoft
Figure 2.3: Pilot’s view from the cockpit. Joysticks, screens and buttons that serve as the user interface of the mech are visible.
Figure 3.1: The mech subdivided into legs (A), upper body (B) and cockpit (C)
Figure 3.2: HTC Vive controller.
Figure 3.3: The steering joystick.
Figure 3.4: Velocity meter. The arrow indicates the current velocity of the mech and the triangle indicates the desired velocity/steering joystick position.
Figure 3.5: Players glove on top of the inactive ACS.
Figure 3.6: Player’s glove with ACS activated and attached.
Figure 3.7: Mech’s left arm replaced with a weapon system.
Figure 3.8: To make hitting targets harder, the player has to aim weapons with open sights similar to real life handguns.
Figure 3.9: Projected path of the mortar shell is drawn with dotted green line while player aims the mortar.
Figure 3.10: Mechwarrior Online. Mech’s body is not visible
Retrieved from: https://venturebeat.com/2015/12/14/mechwarrior-online-celebrates-steam-launch-with-100k-esports-championship/
Figure 3.11: Mech in the hangar.
Figure 3.12: Concept image about mech customization, player holding a paint gun.
Figure 3.13: Mech design version 1.0
Figure 3.14: Mech version 1.0 cabin detail.
Figure 3.15, Aalto in the Game demo testing.
Figure 3.16: Mech design version 2.0.
Figure 3.17: Mech version 2.0 cabin detail.
Figure 3.18: Mech version 2.0 cabin open detail.
Figure 3.19: Arm structure detail.
Figure 3.20: Cockpit perspective of weapon and shield.
Figure 3.21: Mech design version 3.0.
Figure 3.22: Cabin detail top view.
Figure 3.23: Cabin version 3.0 top view, inside hierarchy areas.

Figure 3.24: Mech version 3.0 cockpit front view.

Figure 3.25: Mech version 3.0 weapon design, from top to bottom: Blaster pistol, Heavy minigun and Railgun.

Figure 3.26: Combined transverse and vertical strophospheres showing the common region for grip hand. Dempster, W., Gabel, W. and Felts, W. (1959). The anthropometry of the manual work space for the seated subject. American Journal of Physical Anthropology, 17(4).

Figure 3.27: Overlapping image of common regions for the cockpit panels.
Introduction
Figure 1.1: MechVR’s protagonist’s mech seen from the front.
“MechVR” is a prototype game that adapts the first-person shooter game genre for virtual reality.

The key idea is to place the player’s character inside a “mech”, a large, bipedal, anthropomorphic armored vehicle, which is a common trope in fiction (tvtropes.org, 2019). Having the character move around and interact with the game environment indirectly in the mech’s cockpit, rather than directly through a character, has several advantages for the game experience, such as reduced dissonance and virtual reality sickness.

This thesis documents the game development process, with a focus on the design challenges of adapting the game mechanics of a first-person shooter to a virtual reality environment. It also outlines the thought processes and design choices that resulted from these challenges. We believe that the resulting prototype game, MechVR, can be used as a basis for developing a complete game in the future.

In essence, our intentions led us to the following research question:

**How might one develop an action game that would best support the translating of first-person shooter game mechanics into a virtual reality environment?**

### 1.1 VR AND THE GAME INDUSTRY

Virtual reality (VR) has been in the minds of people for a long time but most of the implementations of this technology in the early days has been mainly non-interactive experiences similar to watching a movie. Some devices focused on games attempted to create some interactive experiences but sadly none of them had a significant impact in the industry. It was not until recently that virtual reality technology enabled accurate and interesting interactions reviving the interest and opening the door once again for game development in VR.

In the past years the industry has been moving towards new wireless hardware options such as the Oculus Quest, HTC Cosmos, PlayStation VR and others, alongside a more polished experience with higher quality graphics and innovative mechanics and interactions. These significant steps towards a higher quality virtual reality experience is what motivated us to explore and develop game experiences in virtual reality and what led us to work on this project.
MechVR is a virtual reality game for the HTC Vive platform. In the game the player controls a mech via devices and instruments inside the cockpit. The player explores a post-apocalyptic wasteland and combats enemies using the vehicle’s various weapon systems. The vanquished enemies, such as other mechs and armored vehicles yield resources that can be traded for weapons and ammunition. To highlight mech’s power, the environment has plenty of destructible objects such as trees and buildings.

The main focus in developing the game has been to test how the first-person shooter game genre can be adapted to a virtual reality environment by using features from mech games. The game lacks story and content, but the game mechanics and mech’s controls are well defined.
1.3 DIVISION OF WORK

Lead programmer, game design: Riku Erkkilä
Riku Erkkilä is the lead programmer of the project and majority of the source code has been written by him. Erkkilä also has the main responsibility for game design, with Rodriguez also heavily involved in the process.

Art and design: Andrei Rodriguez
Andrei Rodriguez is the lead game artist for the project, in charge of the production of the entirety of the 3d models and 2d assets that compose the mech, weapons, environments and others. Additionally, the design of the mech and the visual effects within the game engine and participation in the game design process with Erkkilä.

Programmer: Tuukka Muukkonen
Tuukka Muukkonen worked for the project as a programmer for about 6 months. His focus was on destructible environment and functionality of the hangar/shop.

Sounds: Juan Carlos Vásquez
Juan Carlos Vásquez designed the custom sound effects for the game.

Thesis
This thesis has been written as a joint effort by Andrei Rodriguez and Riku Erkkilä. While Rodriguez focused on the design of the mech, Erkkilä was responsible for the general game design.

The following sections and their subsections are written by Rodriguez:
1.1 VR and the game industry
2.1 About VR
2.2 Game Definitions
3.4 Mech design
3.5 Applying ergonomics in virtual reality games

Following sections and their subsections are written by Erkkilä:
1.2 Short description of MechVR
2.3 Game Genres
2.4 MDA game analysis framework
3.1 Goals for the Game
3.2 Project timeline
3.3 Main game design challenges

The rest of the chapters are written jointly by both authors.
2.1 ABOUT VR

The concept of virtual reality, as we understand it today, has been a subject of conversation since the 1950s. The dream of experiencing new worlds in an immersive way has inspired many people throughout recent history, one of the earliest examples of VR is Morton Heilig’s sensorama machine, an arcade-style theater cabinet that would stimulate all senses. The goal of the machine was to immerse the viewer into the movie that Heilig shot exclusively for this purpose. Heilig later made the first head mounted display (HMD) called the telesphere mask in 1960. Without any sort of tracking, this display provided stereoscopic 3D and surround sound (Mortonheilig.com, 2019).

Another important step in the development of VR was the headsight, created by two engineers from Philco corporation in 1961 (Sutori.com, 2019). It was developed to be a long distance immersive viewing device for the military, connected to a closed circuit camera to be used in dangerous situations. In 1968, Ivan Sutherland and his student Bob Sproull created the Sword of Damocles (Sutherland, 1968), a VR HMD that was suspended from the ceiling due to the considerable size and weight of the machine. Unlike its predecessors, it was connected to a computer rather than a camera and it was capable of tracking the users position while displaying some primitive graphics of rooms and objects.

In 1992 we find one of the well-known VR projects called The CAVE (Cruz-Neira et al., 1992). The CAVE (cave automatic virtual environment) is a virtual reality environment in which projectors cover the walls with stereoscopic images that, in conjunction with special glasses, immerse people in different visual projections. From the 1990s onwards, we see the rise of commercial applications for VR produced by technology companies such as SEGA and Nintendo. Examples include SEGA’s VR headset from 1993 and Nintendo’s Virtual boy from 1995. Sadly, none of these devices had enough impact on the consumer world to allow the virtual reality technology to flourish.

A turning point for virtual reality devices came in 2012, when Oculus released the Rift. This device rekindled the public’s interest in VR technology and caused game developers to consider VR as a viable mass consumption product. However, it was not until 2016 when HTC released the Vive, which had high quality graphics and positional tracking, that virtual reality became a big topic in the technology world and major platform for game development.
2.1 GAME DEFINITIONS

Despite its presence in our lives, there is a degree of uncertainty around the definition of the term “game” (Arjoranta, 2014). For the purpose of this thesis, we decided to adhere to Greg Costikyan, who defines games as “an interactive structure of endogenous meaning that requires players to struggle toward a goal” (Costikyan, 2002). This definition summarizes some of the key concepts commonly addressed in game research: interactivity, structure, endogenous meaning, struggle and goals. These elements can be found from rudimentary games to recent triple-A releases, and they continue to be present in virtual reality games. Understanding and successfully applying these concepts during the game design process sets a solid foundation for a strong entertainment product.

In MechVR, thanks to virtual reality, interactivity is strongly present throughout the whole game experience. The high immersion capabilities of the VR hardware grant a wide spectrum of design options and challenges that enrich the interactive experience beyond traditional video games. In VR players can grab, press and pull objects in a more instinctive way, drastically reducing the learning curve for basic interactions. Because of this, we could push the boundaries of basic game mechanics in MechVR, and reach a simulation-like experience.

New and interesting game mechanics cannot exist without a proper structure. The cohesive relationship between basic rules, context, motivation and challenges allows designers to have a solid foundation where the novel mechanics can bloom. In MechVR the basic structure of the game revolves around the process of controlling the mech’s systems. The core mechanics of the game revolve around the task of piloting a complex machine with different systems that, depending on the situation, present varying degrees of challenge in the game. Every single aspect of the game was designed around this core concept, from the design of the world and environments to the combat situations and enemy behaviours.

Costikyan defines meaning as something endogenous, which is defined by the Merriam-Webster dictionary as “caused by factors inside the organism or system.” “The meaning is caused by the structure but it grows out of it” (Costikyan, 2002). For MechVR, meaning is composed by the world’s structure and the game’s rules and mechanics. However, when players immerse themselves into virtual reality, this meaning becomes even stronger. Because of the high level of immersion and meaning, every action becomes more important and every success more rewarding.

“All the best games are easy to learn and difficult to master. They should reward the first quarter and the hundredth.” (Nolan Bushnell, 1971), with this famous quote in
mind we set to design the challenges for our game, and as mentioned before, our
game experience orbits around the core design concept of piloting a convoluted
machine such as a mech, we designed the challenge to emerge from operating
multiple control panels inside the cockpit effectively. Each of these panels has a set
of analog controls and depending on the situation the adequate and timely use of
these instruments will grant a positive and rewarding outcome or lead to failure.

2.2 GAME GENRES

According to Apperley (2006), video games are so varied that they cannot be
regarded as a consistent medium. It is widely thought, however, that there are
established game genres into which games can be categorized. Our view is that ana-
lyzing a video game should start by considering which of these subsets it resembles
and comparing its features to the archetypes of its genre.

Even though game genres are not explicitly defined anywhere, “established game
genres”, in the language of Apperley, exist. Junnila (2007) states that since some
genres can be found from many different classification systems, some kind of
collective, fuzzy genre system exists. We think MechVR is a cross-breed between
two widely used, yet ill-defined game genres: a first-person shooter and a (vehicle)
simulation game.

2.2.1 MechVR as a first-person shooter game

The first-person shooter (FPS) genre is commonly considered an action video game,
which is played from the point of view of the protagonist. FPS games typically map
the gamer’s movements and provide a view of what an actual person would see and
do in the game (Techopedia.com, 2019).

The first archetypical first person shooter is arguably Wolfenstein 3D (id Software,
1992). Wolfenstein 3D was predated by several video games technically fitting the
broad definition of the genre, but it was the first title to have the key characteristics
common to contemporary FPS games such as Far Cry 5 (Ubisoft, 2018). We consider
these characteristics as action shown from the viewpoint of the protagonist moving
in a 3d environment, emphasis on combat, shooting enemies with ranged weapons
while avoiding getting shot and collecting items that aid combat (such as ammuni-
tion and health packs).
Since we strived for FPS-like gameplay, MechVR employs these characteristics. The mech theme fits fairly well in the genre. For example, in Titanfall 2 (Electronic Arts, 2016) the protagonist also pilots a mech in long sequences of the gameplay. The key difference to typical games of the genre is the realistic, mechanical interface, which controls the two main game mechanics – shooting and moving. In this regard, MechVR can be considered a vehicle simulator game.
2.2.2 MechVR as a vehicle simulation game

Vehicle simulation games are a genre of video games which attempt to provide the player with a realistic interpretation of operating various kinds of vehicles. This includes automobiles, aircraft, watercraft, spacecraft, military vehicles, and a variety of other vehicles. The main challenge is to master driving and steering the vehicle from the perspective of the pilot or driver, with most games adding another challenge such as racing or fighting rival vehicles (Wikipedia, 2019).

In a first person shooter game, the player controls the protagonist who moves and fights in the game world. In MechVR, the anthropomorphic mech does the moving and fighting, while the player controls it indirectly through the mech pilot. One of the goals for the project was to give the player the experience of what piloting a mech would feel like. A large part of the development process was spent on imagining and designing a user interface for a pilot controlling the imaginary vehicle. Learning how to use this interface and operate the mech is part of the game’s challenge. For these reasons as well the game can also be considered a vehicle simulator game.

Figure 2.3: Pilot’s view from the cockpit. Joysticks, screens and buttons that serve as the user interface of the mech are visible.
2.3 MDA GAME ANALYSIS FRAMEWORK

2.3.1 Description of MDA framework

We use the MDA framework as a lens through which we review the design process of MechVR. The MDA framework, which stands for Mechanics, Dynamics, and Aesthetics, is a formal approach for analyzing games. Its authors, Hunicke, LeBlanc and Zubek, state that it “helps clarify and strengthen the iterative processes of developers, scholars and researchers alike, making it easier for all parties to decompose, study and design a broad class of game designs and game artifacts.” The MDA framework breaks game design into three components: mechanics, dynamics and aesthetics. Mechanics describes the hard-wired components of the game, the rules and the data. Dynamics describes the interaction between the player and the mechanics. Aesthetics describes the emotions caused in the player when they are playing the game (Hunicke, 2004).

2.3.2 MechVR viewed through the lens of the MDA framework

Making a game “fun to play” is often the stated goal for game design. In the MDA framework, “fun” is considered to be a composite of aesthetics, different emotional responses that the game causes in players. Hunicke et al. (2004) also provide an example taxonomy for these aesthetics: sensation, fantasy, narrative, challenge, fellowship, discovery, expression and submission.

When the development of MechVR started, VR had only just become a viable mainstream platform and no major VR mech game had yet been published. We wanted to use the capabilities of VR to explore how piloting a mech would feel like, following the definition of aesthetics of the MDA framework. We felt sensation and fantasy to be the most crucial aesthetics of the game. In striving for these aesthetics, we made the mech’s user interface include displays, joysticks and switches that players could operate in the cockpit. We also designed controls to translate the players’ arm movements onto the mech and created an environment with breakable everyday objects such as buildings and cars.

We regarded challenge as another key aesthetic. Typically if a game is either too easy or too difficult, the feeling of challenge disappears and, with it, the fun. In MechVR, there are two major sources of challenge. The first one is typical for simulator games: the challenge of operating a complex system and the feeling of mastery when this is done successfully. The second one is common to first-person shooters: the challenge of combat.
The aesthetics rise from the dynamics and mechanics. Järvinen (2009) defines game mechanics differently from Hunicke, arguing that game mechanics are the means that the game system affords its players, in order to pursue the goals stated in the rule set. MechVR is a prototype game and lacks long term objectives. For the sake of this analysis, we consider defeating enemies in combat and survival the goal of the game. To achieve this goal, players operate the instruments in the mech’s cockpit. Resulting from this goal, the most important secondary mechanics are: moving the mech, turning the mech, rotating the upper body of the mech, moving the mech’s arm, as well as selecting, aiming and firing the weapon systems.

As the game prototype lacks content, the dynamics that arise from the mechanics are somewhat limited. While conducting playtesting, dynamics typical to the FPS genre such as using the environment to avoid enemy fire could be observed. However, we also identified a notable difference. In FPS games, the players typically move while they shoot. In MechVR, we observed this dynamic much less. For the sake of challenge, we made aiming the primary weapons difficult (details in section 3.3.4). During playtesting players appeared to have difficulties with simultaneous aiming and moving. This pushed players to stop the mech in order to hit the targets effectively.
Post Mortem
This chapter is a post-mortem review of the development of MechVR. We start from our objectives and then describe the timeline and major milestones of the project. The main focuses of this chapter are the key game design challenges and the design of the mech itself.

### 3.1 GOALS FOR THE GAME

Our game was born from the realization that VR would provide an interesting environment for a mech game. In our initial discussions we also realized that by placing the player inside a mech, many of the problems affecting FPS games in VR could be mitigated. Based on these insights, we set up our goals for the game, as follows:

1. Simulate the experience of piloting a mech
2. FPS-like gameplay
3. Explore the possibilities of VR for the mech game genre

### 3.2 PROJECT TIMELINE

The MechVR project was started in December 2016 when Rodriguez and Erkkilä constructed a proof-of-concept prototype of the game.

The first public demo was shown at the Aalto in the Game event on April 10th 2017. At that time, the game already had most of its basic functions built. Players could pilot a mech and shoot at floating targets. We were worried that the scope of the game would be too large for such a small team, but encouraged by positive feedback received during the event, we decided to continue development. At this point, we were aiming to produce a multi-player online team vs. team game. We believed this would require less content to be fun to play than a single-player game.

In May 2017, we founded our company, TinkerSquid Oy, to host the development of the game. We applied to the NordicVR incubator programme for funding, but were eventually rejected in the second round. During this process, however, we received advice to reconsider the multi-player approach, since the VR player base was considered too small. Furthermore, the sales of HTC Vive, our primary platform, proved to be disappointingly low at the time.
A demo of our game was also shown at an International Game Developers Association (IGDA) event in November 2017. At this time, there were doubts all around whether VR gaming would become popular enough to support serious game development. We failed to draw the correct conclusions and pressed on developing a self-funded single-player game. We struggled to devise a game that would be interesting to play without much content. Our somewhat misguided solution was to place the game events in a randomly generated world so the experience would be different with each replay.

In March 2018, a year after the first application, our second application to NordicVR was rejected. After 16 months of development, we had developed a somewhat playable game in which players could wander around in a randomly generated world, fight enemies, collect resources and upgrade the mech. It was, however, far from a publishable product. We estimated that one to two years of further development would be required to finish the project, and therefore decided to shelve the project.

### 3.3 MAIN GAME DESIGN CHALLENGES

This section describes the iterative design process of the game, focusing on the key design challenges.

#### 3.3.1 Point of view: Pilot vs. Mech

In MechVR, the player controls a single mech. There are video games, for example, BattleTech (Paradox Interactive, 2018), in which the player has a fleet of mechs. These games are typically strategy games and require a birds-eye-view to keep the player informed of the situation on the battlefield. Since one of the set goals for the game was to simulate the experience of driving a mech we promptly dismissed this approach.

This led to our first major design choice: What would be the player’s point of view in controlling this single mech? We explored two options of how players would experience the game world, one as a mech pilot and one as a mech. We constructed simple prototypes of both options and had them playtested with a small set of volunteers. Below are the descriptions of these tests and the conclusions we drew from them.
A) Player as the pilot

In the majority of games mechs are piloted by human operators. Having the player experience the game as a pilot controlling the mech was the more obvious option. The player would see the game world through pilot's eyes and operate the mech using the controls in the cockpit.

Findings:

(+ ) The user interfaces of the game can be placed naturally in the devices of the virtual cockpit.
(+ ) Character and player are both sitting still, resulting in reduced dissonance.
(- ) Playing as the pilot, rather than the mech, reduces the power fantasy of the player.
(- ) The camera needs to be rotated with the mech's upper body, not the player's head, as this is a major cause of motion sickness.

B) Player as the mech

Since mechs do not exist in our daily lives, it could be argued that they serve to fuel the power fantasies of the audience. These fantasies can either be about being a mech pilot or, with anthropomorphic mechs, being a mech. For this reason, we also explored the less obvious option of a player experiencing the game world as a mech.

In this prototype, the player sees the world through the “eyes” of the mech. They see the metal body of the mech instead of their own, and the mechs' hands and arms approximate the position of player's hands and arms.

Findings:

(+ ) Proved an interesting experience. Players get the novel feeling of having a mech’s body.
(+ ) Less motion sickness from turning. The mech can be turned so that it follows the player’s gaze. This does away with rotating the camera independently from the player, which is a major source of motion sickness in VR.
(+ ) There is no cockpit to house the interface buttons and screens. All the controls of the mech need to be directly mapped to the hand-held input devices, and all information needs to be presented in heads-up display (HUD) style or attached to the mech’s weapons.
There is dissonance due to lack of connection between the players and the mech’s legs.

The forward movement of the mech caused more motion sickness in the player.

Both options, player as pilot and player as mech, seemed viable. We decided to go with the player being the pilot in the early phases of the development. We felt that this option would provide us more flexibility in designing the virtual cockpit. This allowed us to add a myriad of different virtual buttons, knobs, joysticks and screens to control the mech.

3.3.2 Reducing VR-induced sickness

Experiments conducted by Yildirim (2019) have indicated that cybersickness during VR gaming could abase the gaming experience, leading to lower game enjoyment. According to Nichols and Patel (2002), VR-induced sickness is thought to occur as a result of conflicting input to the visual and vestibular senses, possibly explained by sensory conflict theory. In VR, one such conflict results from situations where movement is achieved via a hand-held input device. The visual system is presented with simulated movement, but the vestibular system registers a static position of the participant.

We suspect VR-induced sickness is a key reason why FPS games, popular on desktop environments, are so scarce on VR platforms. Since, typically, the gameplay in FPS games requires fast movement and turning of the player character, this genre is especially prone to causing VR-induced sickness.

We realised early in the project that a mech game would have three major natural benefits in reducing VR-induced sickness, while still achieving FPS-like gameplay. The first of these is that the cockpit forms a static frame of reference around the player character. This has been shown to reduce motion sickness (Duh, 2001). The second benefit is that since the player character is sitting inside the cockpit instead of running in the game world, there is no dissonance from the player sitting stationary in the real world. The third benefit is that since mechs are large and heavy, it is natural that they walk and turn slowly, again reducing the VR-induced sickness. In regard to gameplay, however, this slow movement caused problems in finding challenge in the game (see section 3.3.4).
FPS and mech simulator games typically have a slight vertical bounce of the camera to give the illusion of bipedal movement. This is a potential source for VR-induced sickness. We prototyped and playtested four different scenarios to counteract this. In all scenarios, the player character, i.e. the camera, was attached to the cockpit. In these prototypes, the mech was divided into three parts according to figure 3.1: legs (A), upper body (B) and cockpit (C). We had two versions of the walking animation for the legs: one had humanlike natural bounce to the hips, while the other had none.
Findings from the different scenarios:

A) Bouncy walking animation, upper body and cockpit bounce with it.
   (+) Strong sense of bipedal motion in the cockpit
   (+) Walking looks natural from the outside
   (–) Extremely nauseating

B) No bounce in walking animation, upper body and cockpit also have no bounce.
   (+) Less nauseating
   (–) Walking looks unnatural from the outside
   (–) No sense of bipedal motion in the cockpit

C) Bouncy walking animation, but upper body and cockpit do not bounce with it.
   (+) Less nauseating
   (+-) Walking looks tolerable from the outside
   (–) No sense of bipedal motion in the cockpit

D) Legs bounce and upper body bounce, but cockpit does not.
   (+) Less nauseating
   (+) Walking looks satisfying from the outside
   (+) Passable sense of bipedal motion in the cockpit

We felt that scenario D gave a good compromise between a sense of bipedal motion, while not causing excessive nausea. We decided to incorporate scenario D into the game.

3.3.3 Designing the mech controls

A lot of effort was put into designing the controls for operating the mech. In this section, we describe the iterative design process for the most important controls in the game, which are the mech's movement and shooting with the primary and secondary weapons.

HTC Vive, the target platform for our game, has two identical hand-held controllers. With these controllers, the position and orientation of the player's hands can be tracked. The player can also give input with a trigger, grip, menu and system button, and a trackpad in the controllers. There is also a haptic feedback system that can be used to make the controller vibrate.
Our design goal was to control everything in the game by using only the position and orientation of the controllers and the trigger button (indicated by number 7 in figure 3.2). Using the controllers like this is intuitive - players learn to interact with the game world by virtually “touching”. If the other buttons are used, the player needs to be specifically informed about their functions by pop-ups, possibly breaking the immersion. We eventually ended up making an exception for activating the secondary weapon with the grip button (number 8 in the diagram), see section 3.3.4. for details.

**Movement controls**

The mech moves around the game world by walking and its upper body can be rotated in relation to the legs. This means that the movement direction is independent of the direction where the cockpit is facing. In total, we would need three analogue inputs from player:

1. Desired direction of movement, which designates the desired rotation for the mech’s legs.
2. Desired speed of movement.
3. Desired angular velocity for rotation of the upper body.

Since our design goal was to have the controls present in the cockpit, we needed to design a virtual controlling device with three axes. Since the dominant hand of
the player is occupied for aiming and shooting, the controlling device needs to be operated with just one, non-dominant hand. The same hand is also used to operate other, less-frequently used controls of the mech, such as weapon selection, which creates another requirement: the player needs to be able to grab and release the controlling device swiftly.

Through an iterative design process we ended up with a virtual steering joystick with three degrees of freedom, shown in figure 3.3. The device as a whole acts as a traditional joystick pivoting on ball joint A. The desired movement direction and velocity for the mech comes from the direction and amount the joystick is pivoted to. The mech’s upper body rotates freely in relation to its legs. To enable the player to control the orientation of the mech’s upper body, we added an additional degree of freedom to the joystick: a handle rotates on joint B, along the axis shown in the figure. Rotating along the vertical axis would have been more intuitive, but it was less ergonomic.

The steering joystick is operated by the player moving the character’s virtual glove on top of the joystick and pressing the trigger. As the player holds down the trigger, the glove vanishes and the movement of the controller moves the joystick instead of the glove. The mech’s direction of movement is determined by the direction to which the joystick is pushed and speed is determined by the distance the joystick is pushed. In technical terms, the desired velocity vector for the mech is the vector in the XZ plane from the HTC Vive controllers’ initial position when the trigger is pushed to its current position.

Playtesting quickly revealed that players had difficulties operating the joystick controls that we had designed. The movement of the joystick does not instantly translate to the movement of the mech. There is a delay when the legs rotate towards the desired direction or when the walking accelerates or decelerates. Without the delay, however, the movement of the mech would feel flimsy and unrealistic. As a result, the player only gets immediate feedback of the virtual joystick’s position by looking at it. This is
clearly unacceptable, as the player needs to keep his focus on the events in the game environment around the mech. Players found the steering of the mech too difficult as they did not know when the joystick was in neutral position. They also appeared to overdo hand movements required to use the joystick, since they did not know when it was past the extreme position. This worryingly led to player fatigue.

To counteract this issue, we added an additional screen close to the main screen, in the player’s focus. This screen (shown in Figure 3.4) shows the mech’s current velocity vector and the target velocity, which reacts immediately to the joystick position. Playtesting revealed that the new screen considerably mitigated the problems in steering the mech.

![Figure 3.4: Velocity meter. The arrow indicates the current velocity of the mech and the triangle indicates the desired velocity/steering joystick position.](image)

**Arm controls**

Our mech has an anthropomorphic right arm that can be used to hold items such as weapons. We wanted the player to have direct control of the arm, so that when the player moves his hand, the mech’s hand mirrors the movement.

To switch between the default mode in which the player character’s glove operates the controls in the cockpit and the mode, where the glove is used to move the mech’s hand, we created the virtual Arm Control System (ACS). The ACS is an object in the cockpit, which, when touched, activates and attaches to the glove. When the ACS is active, the mech’s hand mirrors the character’s hand position, and pressing the trigger activates the item (e.g. a weapon) in mech’s hand. While playing the game, players typically have the ACS activated at all times, but if they want to operate other devices in the cockpit with their right hand, the ACS can be deactivated by touching its docking point.
We were satisfied with this solution as it conforms with our general design goal to have all controls visible in the cockpit, while still allowing the player to operate the mech’s hand/arm as if their own.

Full control of the arm also allowed us to have a weapon selection mechanic very common in VR shooters: when the player moves the mech’s hand behind its back, the weapon in the hand cycles to the next one in the inventory. There are also buttons in the cockpit to directly select any weapon. When these are pressed, the player loses the control of the arm for a second as the mech autonomously reaches behind its back for the weapon change.

**Secondary weapons**

Controlling of the secondary weapons also went through an iterative design process.

**Iteration 1:** The mech had two similar, anthropomorphic arms. The secondary weapon was held by the non-dominant hand. The player had constant full control over both arms.
(+ ) Having players control both arms constantly and simultaneously made players feel like they were “one with the mech”, catering to the power fantasy aspect of the game.

(– ) The player’s non-dominant hand is needlessly tied up all the time with controlling the seldom used secondary weapon. There cannot be any controls in the cockpit, since the player cannot use them.

**Iteration 2:** The mech’s non-dominant arm was replaced with a secondary weapon system as shown in figure 3.7. It had a dedicated controller system in the cockpit, near the steering joystick. It was a pistol grip shaped joystick mounted to a ball joint. The secondary weapon aimed at the same direction as the controller’s shaft.

(+ ) The player could let go of the steering joystick and quickly operate the secondary weapon.

(– ) The mech stopped when a secondary weapon was operated.

(– ) Most other controllers in the cockpit are activated by placing the character’s glove on them and pressing the trigger. Since the trigger was needed to fire the weapon, the aiming system instead activated just by placing the glove near it. This felt unintuitive and lead to confusion, for example with involuntary firing.

**Iteration 3:** When the touchpad is being pressed on the HTC Vive controller, the player controls the secondary weapon instead of the arm with the primary weapon.

(+ ) It is very fast to switch between primary and secondary weapon.

(+ ) It does not interfere with the movement of the mech.

(+ ) Versatile. The way the controller’s position and buttons are mapped to aiming/ firing can vary by the requirements of the weapon.

(– ) Goes against the general design goal of having all controls in the cockpit.

(– ) Exception for the general design goal of having all interactions happen with the trigger button of the HTC Vive controller.
3.3.4 Finding the challenge

In a typical desktop FPS game aiming is not that difficult as the player has a crosshair that can be quickly and accurately moved with the mouse. A big part of the challenge of these games is the movement, “thinking with your feet”. In our game, fast movement is not possible due to issues with motion sickness and also because a nimble mech would feel unbelievable. This left us with a question: Where does the challenge then come from? Our solution was to make it harder to aim the weapons.

Primary weapon

In the early versions of the game, the player had a crosshair on the screen of the mech that indicated where the main weapon was aiming. This made aiming and hitting targets too easy. We removed the crosshair and added open sights to the main weapons, similar to pistols in the real world. This certainly made hitting things harder, but also created new design challenges. To get the aiming to work, major changes to the structure of the mech were needed. Aiming the weapon with open sights requires the hand in question to employ six degrees of freedom, so the player can align the front and back sights with their view. To achieve this, we had to add a third joint to the arm of the mech. Later, we also added a fourth joint to the “clavicle” of the mech, to get more reach for the arm to help with the aiming (see section 3.3.4).

Figure 3.8: To make hitting targets harder, the player has to aim weapons with open sights similar to real life handguns.
Secondary weapon

During the development process, we also adjusted the difficulty for aiming the secondary weapon, the mortar, in three iterative steps. The mortar shells fly in a ballistic arc.

**Iteration 1:** The shells launch at constant speed. While the mortar is aimed, the shell’s projected path is drawn.

Findings:
- Aiming and firing was easy and intuitive.
- Since the launch velocity of the shells is constant, it is not possible to do trick shots, for example firing a shell with low velocity and high arc in order to hit a close target behind an obstacle.
- Hitting targets was too easy.

![Figure 3.9: Projected path of the mortar shell is drawn with dotted green line while player aims the mortar.](image-url)

**Iteration 2:** Same as iteration 1, but with the projected path only drawn until fixed distance from the mortar.

Findings:
- Hitting targets at close range was as easy as in iteration 1. The shell will hit where the projected path indicator line intersects with the ground.
- Hitting targets at long range was too hard. When the target is far enough that
the projected path indicator cannot reach it, it is very hard to estimate where
the shell will land. This is most likely because as the projected path is drawn on
an empty sky, it is hard to estimate how far its end is in relation to objects on
the ground.

**Iteration 3:** Same as iteration 1, but the launch velocity of shells is not constant.
When the mortar is fired, the launch velocity starts low and is quickly increased
while the trigger is held. While the velocity increases, the projected path is con-
stantly updated. The shell is fired when the player lets go of the trigger.

Findings:
- Impact position of the shell is as easy to predict as in iteration 1.
- With variable launch speed, trick shots are possible.
- The difficulty now comes from timing the release of the trigger at the right
time. The difficulty can be easily adjusted by changing the rate with which the
shell launch speed changes.
- Playtests revealed that this mechanic is confusing for new players as they do
not know that the trigger is supposed to be held down. This resulted in acci-
dentally firing the shells very close by and in some cases not realizing that the
velocity can be altered.

**Iteration 4:** Same as iteration 3, but with an added grace period in firing. If the
trigger is released almost immediately, the mortar does not fire.

Findings:
- Same as in iteration 2 except players were not prone to accidental
  close shots anymore.
- Players typically first tried to release the trigger immediately. They were tipped
  off by the mortar not firing and the projected path flashing quickly on the
screen, and typically quickly realized that the trigger is supposed to be held.

**3.3.5 Tinkering with the mech**

An interesting aspect of a VR mech game is that the player gets a sense of the
dimensions of the mech. On a desktop game, most of the mech’s body is invisible
to the player and they can typically only see the HUD display and perhaps parts of
the cockpit. In MechVR, the player can see and control the arm of the mech and by
looking around see the shoulders and legs of the machine.
We figured, that since the player can see more of the mech, they would also be more inclined to customize it. For these purposes, we created a hangar mode for the game in which the player can view the mech from the outside and move around it in a lift. From this viewpoint, the player can customize the mech, adding new parts and painting it with a spray gun. The hangar was only implemented partially – it was used as a kind of shop, where resources could be traded with weapons and ammunition.
3.4 MECH DESIGN

3.4.1 General design based on function

The mech is the main pillar of the entire game experience and its design was a key part of the project, going through many tests, iterations and redesigns. All of this provided us a clear perspective of the essential characteristics needed for a competent VR experience for controlling such machines.

VR grants a significant level of immersion for users mainly thanks to the quality of the hardware used (HTC Vive, Oculus Rift, Gear VR). Nonetheless, the software plays an equally significant part in a successful VR experience. To achieve an acceptable level of immersion, we decided to focus our efforts on creating a believable design that, even if set in a fantasy world, would have a logical functioning structure for a machine of this sort.

This required us to look at the design from a functionality standpoint more than just an aesthetic one. The process was not simple due to the continuous changes made as a reaction to results from internal testing and user feedback. After establishing the essential functions that the mech should have for the core gameplay, we went through extensive research of mechanical references in search of an interesting
general look for the machine. Our aim was to achieve visual elements that would fit both functionally as well as aesthetically.

The most relevant characteristics that we chose were mechanical systems, such as pivot, fork, hinge and ball joints, hydraulic pistons and lines. For the external details and appearance we chose surface details, like riveted armor plates similar to those found in tanks and aircraft. For more specific details we picked some industrial and heavy machinery features, such as metal and rubber hoses and pipes, rugged structural supports and beams, grill vents and exhaust pipes, among others.

### 3.4.2 Evolution of the design

The design of the mech started mostly from existing references as an easy way to test most of the game concept and mechanics, without venturing into high-complexity models. The design process underwent several modifications and redesigns as the result of many internal and user tests as well as game design changes made during development.

The evolution of the design can be reduced to four main phases, as follows:

**Proof of concept (Version 0)**

The primary requirement for the project was to have a simple starting structure that could give us enough functionality to do a proof-of-concept demo. Our initial idea was to test the basic interactions that the player would have with the machine and the environment in VR. This encompassed all essential motion, rotation and displacement, as well as aiming and shooting at targets located in the environment.

As a starting point we decided to take cues from existing designs found in well-known games like Battletech (1984) and Warhammer (1983), and movies like Robocop (1987) and Avatar (2009), among others. The design features a massive body volume where the player cabin is located, sometimes displaying extra attachments, such as rocket pods or energy sources. This was supported by a double-jointed bipedal locomotion system and symmetrical arm-mounted weapons, located at the sides of the cabin. In some cases, we observe the pilot constantly controlling the arms to aim and walking in the direction the mech is facing.

This approach uses traditional VR interactions seen in other popular experiences, placing the player in the middle of a small area where they can walk around and
interact with different elements. The voluminous body gave us enough space to move freely while traversing the VR world. Our gray-boxed 3d mesh featured a closed cabin with some minimal ceiling height and a wide window in the front, accompanied by some basic movement controls mapped to the touchpad of the Vive controllers.

Initial tests proved that standing up while controlling a moving vehicle in VR tends to create vection, also known as motion or simulator sickness (Hettinger et al., 1990), and balance reduction for many users. This problem led us to instead make the players sit down while controlling the mech. Sitting proved highly effective in reducing motion sickness effects and concurrently increasing immersion during gameplay, thanks to the familiar cognitive models established by modern vehicles.

Since we wanted to grant the players continuous aiming capabilities, the arms were constantly reacting to the movement of the player’s hands using a proprietary IK system developed in-engine. To accomplish this, the whole arm structure was initially supported by a significantly large ball joint for the shoulder and a smaller one for the elbow. The forearm was equipped with a gray-boxed cannon following the ascribed concept.

This configuration allowed for enough rotation during movement tests. However, we noticed that having the mechanical arm move at the same speed as the player hand was not believable. Considering the weight that such a structure would have, the motion speed had to be reduced in comparison. To compensate for the speed difference, a prediction system was implemented, which displayed two crosshairs in front of the player. One tracked the accurate position of the player (controller) aim, while the other displayed the real position of the mech’s arm. For the player to successfully hit the target, both crosshairs had to be in the same position.

The leg structure featured a bipedal triple-jointed system, supported by a ball joint for the hips and three hinge joints for the knees and ankles. The whole structure was rigged and pre-animated in 3ds Max before it was brought into the game engine. The legs, alongside the cabin, fully depended on the orientation vector during displacement. This quickly presented many problems during combat situations, due to the lack of strafing or sideways displacement, which made it impossible to aim or shoot in one direction while walking the opposite way. To resolve this issue, we modified the system to be independent from the cabin orientation. It continued to follow the walking orientation vector, but the cabin rotation was mapped to the controls as a motion input instead.

After internal tests and prototyping, this simple structure provided an acceptable starting point for the proof-of-concept demo. The result was a basic mesh,
controlled mainly by the motion of the arms and touch input from the controllers while the player is sitting down.

**Cabin redesign (Version 1.0)**

During early phases of development, we were interested in acquiring greater user feedback on the game concept, which led us to participate in the Aalto in the Game event, on April 10th 2017. For the upcoming event, we decided to create a first 3d mesh design that replaced the primitive graphics that were used for version 0.

Structurally the most significant change was made to the proportions of the cabin, as the player would no longer be standing up. This meant a significant reduction in the height required for the player to move, resulting in a cuboid-shaped room with a single seat located in the center. Two button panels, located in front of it, were used as placeholder position markers, since the initial prototype did not feature any internal system interactions. The exterior of the cabin featured two purely decorative ventilation ports on each side (see Figure 3.14), which also served as the attachment point for the shoulders.

![Figure 3.13: Mech design version 1.0](image1)

![Figure 3.14: Mech version 1.0 cabin detail.](image2)
After internal testing, we determined that the cabin space for the player seemed excessive and created a feeling of emptiness, which ultimately made the mech feel like a small room with legs.

Version 1.0 had the same arm structure as in version 0, now with improved 3d graphics. These featured a simple arm and a forearm chassis connected by smaller ball joints and sockets. On top of the chassis, three individual meshes of armor for the shoulder, elbow and forearm were added to gather results on mesh complexity and detail in VR.

During initial arm tests, the arms behaved mostly like in version 0 within non-demanding gameplay situations and simple combat. However, due to the cabin being slightly wider because of the vents and shoulder attachment, the horizontal adduction range of the arm created serious mesh clipping issues with the cabin, greatly reducing the horizontal range during complex combat situations.

The support structure featured the same independent bipedal system as version 0. For the final design, the leg pieces were modelled in detail, including the attachment and pivot points for the hinge joints. Smaller details, such as wires and cables, were added to the system through rigging.

For our first user test during the Aalto in the Game event, our focus was to see how people would control the mech in an open environment with no predefined goal, and how fast they could adapt to this experience. Therefore, we created an open shooting range environment with some bullseye drones hovering over the area. The 3d meshes of the mech alongside the drones were UV mapped and textured for the event.

Testing the demo at the event resulted in a very positive response from users in regards to the general experience. Most of them did not experience any discomfort or motion sickness during testing, which suggested the sitting position was effective for this purpose. Nonetheless, after some time, minimal discomfort still appeared for a few testers.

Regarding the movement and shooting interactions, we noticed that players were able to control the mech’s movements and shoot separately with no difficulty. Although in more complex situations, requiring both actions simultaneously, testers had trouble effectively controlling the mech. We also noticed that aiming with both arms at the same target was easy, but due to the reduction of arm adduction, targets that were located alongside the mech had to be focused only by one arm, which rendered the opposite arm redundant.
New concept design (Version 2.0)

The positive results and feedback obtained from the Aalto in the Game event, proved the appeal of the main experience and mechanics, while at the same time highlighting the design flaws of version 1.0.
Since our goal was to develop our own design and move away from our initial reference, we decided to completely redesign the whole structure, systems and parts of the mech, based on the test results of previous versions.

The first major changes concerned the size of the mech. We wanted the players to experience it as a complex and powerful machine, therefore we increased the total height from 3 meters to 7.3 meters. This strengthened the perception of the general capabilities of the machine and presented players a new enjoyable perspective of the game world.

During our development we considered a multitude of game elements to introduce to the experience. One of the most important gameplay features we decided to add was the customization of the mech in the hangar. This decision required the design of the mech to be structurally elaborate enough to be easily armed and disarmed without breaking the immersion.

Based on our experience from version 1.0, we designed all the armor pieces to be modular, fully independent and thus removable from the chassis. This meant integrating more complex mechanical systems like pistons, cables and wires into the chassis design. For the armor pieces we created a double-sided 3d mesh, as each of them would potentially be used as an interactive game object by the players during customization.

Version 1.0 had some interior design issues in the cockpit area, alongside a big problem with mesh clipping caused by the wide cuboid shape of the cabin and the joint used on the shoulder. We therefore put our efforts into re-designing the entire cabin from the inside out.

Based on the test results of version 1.0, we decided to try a radically new approach for the cabin shape. Version 2.0 of the mech was characterized by a more egg-like form compared to its predecessor. This shape granted enough room for arm mobility without the issue of having an exaggerated volume.

Following our goal of achieving a heavy look for the armor of the mech, the most radical change was a frontal main access door, which replaced the prominent window frame from version 1.0. To create the main access to the cabin, the front of the mech now featured four moving panels operated by hinge joints and piston-like
objects. We opted for a believable system operation during opening and closing, since the players would be able to see these systems from inside the cabin and potentially from outside when entering the mech.

In the cabin design phase, we focused heavily on giving players a sense of the walking motion of the mech to increase players’ immersion. During the initial tests, we attached the cabin to the hip controller in the rig, to obtain the desired motion during walking. The results were not positive since this method created an almost continuous motion for the players, which rapidly induced motion sickness. To solve this problem, we redesigned the cabin structure to be completely separated from the player chair (cockpit). Furthermore, we attached the exterior structure of the cabin to the hip controller, which created the desired motion during walking without affecting the player position at all.

Inside the cabin, a detailed cockpit structure was designed to accommodate the player and the main control panels featured for this new version of the game. The player chair was placed in the center of a structure resembling a C-pillar, which in automotive design refers to the rearmost structural pillar behind the rear door of a car. A touchpad system was placed alongside the chair, which housed the mech’s hand controls, a secondary set of button panels, one dedicated panel for the PDS (power distribution system) and others for weapon selection, radar use and startup sequence.

To account for the new features including opening and closing armor plates, moving button panels and others, we added systems like hydraulics, power reactors, gyroscopic stabilizers, radar and communications devices into the chassis. These systems complemented the machine’s design from an aesthetic point and presented a high potential for gameplay situations.

The arms of the mech also underwent a drastic redesign, as version 1.0 provided enough basic functionality but still presented many problems. For version 2.0 we decided on a more human-like structure, which for the first time included the hand.

We decided to test the uses, benefits and problems that a hand could provide. The first thing we noticed was that the hand provided a completely new experience for
the players when using the arms of the mech. The use switched from a completely detached control scheme that just followed a marker, to a more accurate representation of the player movements. This improved the overall aiming accuracy and movement precision of the mech. The wrist rotation ranges and sideways aiming was also slightly improved thanks to the extra rotation from the new ball joint.

The new arm structure also provided us with a variety of new interactions, such as hand gestures and grabbing capabilities, which opened up new gameplay possibilities aside from combat. Examples included extracting valuable objects from buildings and confined spaces, secondary missions centered around picking up delicate objects without crushing them, or removing obstacles, such as fallen trees from the mech’s path.

The new arm structure also necessitated a complete redesign of the weapon usage. It went from being fully integrated on the arm structure, to being an independent object held on the hand and carried attached to the mech. This new layout allowed us to create multiple variations for the weapons which now could easily be swapped depending on the situation. During testing we experimented with pistol-like designs, automatic SMG-like designs and even long range options. These proved to be more effective, since their usage became more realistic. This increased the handling of the weapons, but slightly reduced the player’s perception of controlling a complex mechanical system.

During initial tests of the new two-hand design, we decided to try various configurations of weapons during combat, such as two-handed heavy weapons, dual-wielding...
weapons and even different weapons on each hand. During the initial prototyping, however, we discarded the two-handed heavy weapons, because of the immersion breaking issues it presented. Considering that the player would have constant control over the movement of the arms, we found ourselves with two possible options: automatically attaching the hands to the weapon grips and modifying the aiming system, or maintaining the current aiming system and designing the heavy weapons to work as a single-handed grip. The first option took control away from the player to accurately position the hands on the weapons, which resulted in confusion and uncertainty from the player. Potentially, this option could render better results in simulating the weight of the weapons, but further testing would be required. The second option presented a smoother experience since it did not change the aiming system, but it created a new challenge in the weapon design. The perception of weight was similarly affected by the natural movement capabilities of the arm based on player motion. Ultimately, we favored the second option, as it presented a less aggressive transition for the player when switching from the other weapons.

Figure 3.20: Cockpit perspective of weapon and shield.

Dual-wielding resulted in players predominantly using the weapon closest to the target. A different weapon on each hand resulted in a similar behaviour, however, players tended to momentarily forget which type of gun was being held on which hand. This variation led us to also test a shield device for the opposite hand. During
testing the initial results were positive, with players using the shield during simple combat situations. However, during further tests with new users, we noticed that at the beginning of the game the players consciously used the shield located in the left hand, but as the game progressed they dismissed the shield, focusing mostly on the hand with the weapon.

After further testing of version 2.0, we concluded that this design improved on the perception of weight and size for this kind of vehicle, which resulted in a more immersive experience during gameplay. However, we were not fully satisfied with the general aesthetic. The clean and “organic” design felt too minimalistic and futuristic, and did not fit our overall post-apocalyptic design idea.

Increasing the height of the mech drastically improved the perspective that players experience from the game world. It granted a more empowering and clear overview of the game level and its elements. The control systems inside the cabin, however, would require a better layout and design for optimal usability during gameplay.

Due to the high complexity of the design, the 3d model of the mech reached around 60,000 polygons in version 2.0. This represents a considerable amount of geometry and as such started creating performance issues inside the engine. These performance issues were further augmented by the shader. The toon shader used in the scene created an outline rendering by duplicating the original mesh, which effectively doubled the polygon complexity of the model. An alternative was required.

During the gameplay tests we concluded that players do not engage in gameplay situations using their non-dominant hand (left in the majority of cases tested) and that players focus on attacking the target and controlling the movement of the mech to avoid incoming attacks.

**Final version (Version 3.0)**

The major improvements done during version 2.0 cemented the foundation for a strong mech structure, however, the performance issue was a pressing problem that required an immediate solution.

Lowering the mesh complexity of the mech model was the most relevant issue during this stage. To improve this, we started a complete redesign of the core elements of the mech, starting from the cabin and cockpit.
The general structure of the cabin remained the same as version 2.0, with a complete redesign of the cabin exterior pieces in order to reduce the mesh complexity and align them with the desired aesthetic for the game. For the basics of the structure, a fully modular chassis was designed to hold the cockpit structure, the mech reactor, a hydraulic system and two side panels holding 10 tracking missiles. The whole upper body of the robot was changed to a more robust and angular design, taken from modern-day armored vehicles such as BTRs (Armored transport vehicles), tanks, etc.

The main shape of the cabin became more arrow-like in the front, which drastically improved the arm adduction while aiming to the opposite side during gameplay. Additionally, the volume of the cabin was reduced by 20% from the previous version. The clean and angular surface design considerably reduced the polycount for the armor plates of the cabin, while maintaining a solid and consistent look.
For the interface, we also decided to move away from the “futuristic” design and towards a more analog and mechanical style, that better fitted our desired aesthetic. We wanted the design of the interface to fit each of the controls in the cabin based on its functionality, all within the logic of the game’s universe. This led us to find inspiration in control panel of airplanes, nuclear plants, war ships and others. Due to this, and the reduction of the available space inside the cabin, the interior sustained a drastic redesign, improving on the general form, positioning and functionality of each one of the control panels and instruments.

During the redesign of version 3.0 we established a hierarchical structure for all the controls based on their functionality. This allowed us to position the different control panels and information screens logically inside the cockpit. The list was structured as follows:
<table>
<thead>
<tr>
<th>Tier</th>
<th>Item</th>
<th>Type</th>
<th>Relevance</th>
<th>Area/Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tier One</td>
<td>Steering joystick</td>
<td>Control</td>
<td>High</td>
<td>A1</td>
</tr>
<tr>
<td></td>
<td>Hand control system</td>
<td>Control</td>
<td>High</td>
<td>A1</td>
</tr>
<tr>
<td></td>
<td>Weapon switch panel</td>
<td>Button panel</td>
<td>High</td>
<td>B1</td>
</tr>
<tr>
<td></td>
<td>Radar controls</td>
<td>Button panel</td>
<td>High</td>
<td>B1</td>
</tr>
<tr>
<td>Tier Two</td>
<td>Map / Navigation controls</td>
<td>Button panel</td>
<td>Medium</td>
<td>A2</td>
</tr>
<tr>
<td></td>
<td>Energy management system</td>
<td>Button panel</td>
<td>Medium</td>
<td>A2</td>
</tr>
<tr>
<td></td>
<td>Radar &amp; Navigation screen</td>
<td>Display</td>
<td>Medium</td>
<td>C2</td>
</tr>
<tr>
<td></td>
<td>Weapon status &amp; ammunition screen</td>
<td>Display</td>
<td>Medium</td>
<td>B1</td>
</tr>
<tr>
<td>Tier Three</td>
<td>Cabin door controls</td>
<td>Lever</td>
<td>Low</td>
<td>D1</td>
</tr>
<tr>
<td></td>
<td>Startup instruments</td>
<td>Button panels</td>
<td>Low</td>
<td>D1</td>
</tr>
</tbody>
</table>

Table 3.1: Mech version 3.0 cabin hierarchy areas

Figure 3.23: Cabin version 3.0 top view, inside hierarchy areas
The basic control scheme for the mech was redesigned alongside the cabin, based on the feedback and testing results obtained from version 2.0. Considering most of the test users had trouble controlling the movement of the robot using the touchpad on the controllers in combat situations, and often ignored the secondary weapon or shield attached to the left hand, we decided to remove the left arm and replace it with a high damage weapon controlled by the existing right-hand controls. For movement we created the steering joystick which was discussed previously in chapter 3.3.3.

In an internal test session we noticed that during exploration and parts of combat, the general movement of the mech was slower than optimal. Considering that the speed of the mech was intentionally low to reduce motion sickness, we decided to test different ways of giving the player more mobility during these situations. To increase the movement capabilities of the mech we implemented burst movement actions that allowed the mech to dash in different directions depending on the joystick input. This action furthermore allowed the mech to boost up into the air, improving its vertical mobility.

To maintain consistency with the internal features and functionality that were added to the mech, we needed to design the external features of the robot accordingly. A complete structure was designed and implemented to the back of the robot that incorporated two sets of boosters and wings. These were later redesigned to serve as weapon holding structures.

Continuing the visual redesign of the mech, we decided to also redesign the weapons to fit the new aesthetic.
The complete structure of the lower body of the mech remained the same as version 2.0, but with a considerable optimization on the mesh to reduce the polycount. The armor plates for the legs were redesigned to match the new design aesthetic.

For the walk and movement animations a whole new rig system was implemented seeking to polish and ground the robot’s feet during the walk animation. A more
The complete set of animations was done for the various world interactions, including side steps, forward dash, hover, etc.

This redesign and optimization process not only led to a more appealing design and more balanced proportions of the mech, but reduced the polycount by 30,000 polygons (50%) compared to the initial version. This considerably improved performance inside the game engine and during gameplay.

### 3.4.3 Multi-system character design

During the development of the mech structure, we concluded that its complexity went beyond traditional game characters due to the potential variety of gameplay actions within the game. The mech had to offer a wide range of movement options during combat and exploration, which meant that a traditional bone animation system would not be enough. To solve this issue, we split the mech into four major structural groups, depending on their function: the cabin, the cockpit, the arm and the legs.

The cabin required a more specific animation system. Starting out, the main entrance and reactor cover door were handled with a traditional animation system, since it only required two different states: open and closed. To translate the walking motion into the player's view, the cabin was linked to the main hip controller of the leg rig system. Translating this motion directly from the legs to the cabin resulted in a drastic and repetitive change in height during walking, which caused considerable discomfort after a few minutes of gameplay. To reduce this effect, we used a script that allowed us to control its intensity depending on the situation (walking, running, fighting, etc.).

The cabin parts were organized in a group hierarchy under a cylindrical form connecting with the hips. This provided a 360° rotation range for the cabin, while staying independent from the orientation of the legs. The cockpit remained as a part of the hierarchy, being affected only by the movement and fully excluded from the hip motion.

The arm was the most complex system to develop. It was composed of a hierarchy of parts that structured the collar bone, shoulder, arm and forearm, followed by the hand. Since the movement of the arm is controlled by a translation of the player's hand motions into the mech's hand position, to mimic its behavior, we developed a proprietary inverse kinematic system inside Unity. This system controlled the movement of each part of the hierarchy and joints dynamically using scripts. The
hand was made using a traditional bone and animation system that was triggered by button presses to pose the fingers for different actions. The functionality of the secondary heavy weapon was fully implemented using scripts that controlled its rotation based on the player’s aim (see chapter 3.3.4).

The basic functionality the legs have within the game is to represent the action of walking through the environment. Considering this, we felt a traditional IK rig with a pre-made set of animations sufficed. Since the cabin was detached from the design of the leg structure, we decided that, during directional changes in walking, the legs would rotate using scripts and its orientation displayed by the velocity indicator (see 3.3.3). This simplified the overall animation process to a few basic animations. Using this mixed development system, we achieved the complex functionality that we needed for our mech experience to feel consistent and usable during gameplay.

3.5 APPLYING ERGONOMICS IN VIRTUAL REALITY GAMES

The study of game ergonomics has been limited to whatever physical hardware interface is placed between the player and the digital content. Designers and engineers continuously work to perfect the ergonomics for different controllers, joy-sticks, keyboards and mice. With virtual reality slowly becoming a prominent topic in the video game industry, new hardware has emerged, presenting a fresh take on hand-held devices and head mounted displays (HMD).

However, there is one more key element that needs to be considered, which is the whole human body. Neglecting the basic ergonomic principles during the design of a VR experience could potentially have a negative impact, more specifically on those experiences that rely on the players sitting down for long periods of time.

3.5.1 Human-centered VR design (Industrial design in VR)

VR experiences and games are centered around a person interacting with a specific physical interface during a set period of time in order to achieve a result or perform an action. This brings design for VR to the level of industrial design, wherein the effects and limitations of the human body play an important role and should be considered during the design phase.
Virtual reality technology has improved over the last two decades, which has led experts to take a deeper look into the effects and impact of this technology on its users. Many empirical ergonomic studies have been conducted on the effects of VR and its impact on the future development of the technology (Nichols, 2002). Others have focused on the appropriateness and usability of hardware and the effects of participation in virtual environments (Wilson, 1999). Both are common perspectives that experts in the field of ergonomics have used to study and analyze VR in more depth. However, from a game design perspective, we can apply these ergonomic principles while designing virtual workspaces, vehicle interiors or other digital environments that resemble real life ones.

3.5.2 Designing the cabin and controls

Studies in the field of ergonomics and anthropometrics can be found as far back as 1857 (BakkerElkhuizen, Work Smart - Feel Good, 2019). Their focus has mainly been on improving work conditions and symptoms. When designing a product and environment that interacts with human beings, a set of ergonomic, anthropometric and biomechanical principles should be considered to achieve a successful interaction, while reducing the potential risk of exhaustion or even injury for the end user. If we understand virtual environments as digital representations of real and imagined environments, we can define the cabin of the mech as a simulation of a vehicle control cabin. As such, it has ergonomic implications on the player, which can be analyzed and designed.

In MechVR, the design of the cabin and controls was one of the core pillars of the development process and therefore the focus of our ergonomic analysis. Our initial consideration was that the player will be in a constant sitting position during the game. As it is impossible for us to ensure that players will have similar seats and postures, we can limit our design study to the reach ranges inside the cabin. Additionally, we wanted to explore how this would impact the design and positions of the various control panels and instruments.

The area surrounding a subject in which the arm can move while sitting down and having a hand grip position, according to ergonomic studies on arm motion ranges and reach, as illustrated in Figure 3.26 (Dempster, Gabel and Felts, 1959). This particular study is relevant since it relates directly to the ergonomic positions the players will occupy in MechVR.

Considering the position and size of the area in question, we determined the locations for the three main tiers of panels and controls (see Table 3.1) should be mainly
in the surrounding area of the torso and the head. For tier one, the steering joystick needed to be inside this ergonomic range, as it is the most important interface element in constant use. The ideal position for it was right next to the player’s left knee, which allowed the player to use this control effectively for longer periods of time without experiencing fatigue.

The second most important element in tier one was the arm control system (ACS). Using the ACS, the player controls the movements of the arm during combat and exploration. Due to its frequent use, the best position was opposite to the steering joystick, next to the player’s right knee. We lowered the position of the ACS in order to prevent accidental interactions during arm rest without compromising the accessibility of the system (see Figure 3.27).

The tier two button panels, such as the energy management system, radar and navigation controls, would be used less frequently during gameplay. Nonetheless, they required easy access, which was the reason why they were placed inside the mid-height area of the reach range. The player would not directly interact with the weapon status and navigation display, which allowed us to position them outside the comfortable reach, while still within the visible range of the player.

For the third tier of controls, the startup and door control instruments ranked at the bottom of the hierarchy, since they are mainly used during the extremely rare
startup sequences. This allowed us to position them inside the reach range above the head height, maintaining proper accessible distance without compromising the players viewing angles and avoiding instrument clutter.

Figure 3.27: Overlapping image of common regions for the cockpit panels.

Basing the mech's cabin control panels on these basic ergonomic principles, we noticed a great reduction in player fatigue during long play sessions, from earlier versions. Additionally, we noticed that pairing this structured layout with vibration feedback on the controllers enhanced the usability of the various panels, allowing players to react faster during gameplay. These considerations could be employed in our further game design. Through the reach range we may determine more challenging positions for specific panels that need to be used during extreme game situations. We might, for example, force the player to temporarily leave the static, up-right sitting position and take on a less sedentary posture.
4.1 LESSONS LEARNED

During the development process of MechVR we iterated through multiple versions of the mech design and the gameplay. Each tested version yielded the same positive result: The concept of controlling a complex machine in a virtual environment is the strongest part of the project. This maintained, and each time rekindled, our motivation to continue working on the project even through difficulties.

One of the toughest parts of the process was our goal of developing, what we call, a “low budget FPS”. We ran into many obstacles that from some perspectives could cripple, or even jeopardize, the development of the concept as we envisioned it. We had to discard game mechanics commonly successful in traditional FPS games, such as quick aiming, fast movements and quick time events, because they could not be translated easily into a virtual environment. Some mechanics we managed to develop, such as climbing, crouching and lying down were causing virtual reality sickness in the testers. We managed to successfully implement actions like jumping and strafing by reducing the speed of the mech’s movements. This worked nicely considering the size and imaginary weight of the mech.

Experiencing a completely new set of challenges caused a substantial increase in production time and development of the project, which leads us to the second most important lesson learned, managing the scope of the project.

When developing the proof-of-concept demo, we planned a vast set of mechanics, gameplay situations and stories that formed a post-apocalyptic world of adventure, full of action and wonder. Unfortunately, each step towards this goal was longer and more troublesome than the last, making the final production too great and complex of a task to achieve for such a small team. Developing the basic game mechanics for a VR game proved to be a longer process compared to those of traditional games, as many of them required thorough testing to mitigate player VR sickness. It was equally time consuming to design the virtual, analog controls used inside the mech, since players had to perceive, recognize and effectively use them in the game. If the form or position of any particular panel, button or lever was unclear, the usability would suffer considerably.

We employed two other talented team members for a short time during production, but ultimately the road towards a polished final product is still long and laborious.

The current state of VR technologies is exciting and troublesome at the same time. Since devices like the Oculus Rift and the HTC Vive came out, the production and consumption of VR has expanded into many new disciplines and markets, while at
the time increasing its market share (Superdataresearch.com, 2019). VR has become a powerful tool in many fields, and is used for learning and training professional staff. It is also a versatile means of presenting various products in an accessible and economic fashion. However, VR technologies still have not reached the mainstream. Effects of this can be seen in particular in the almost non-existent multiplayer communities, that have turned many PvP games into desolate experiences.

The HTC Vive is an excellent product to experience VR, but the comparatively high price, the space requirements and the need for a high-performance computer has significantly limited its market reach. Recently, most VR hardware companies have started developing wireless solutions that will finally eliminate the need for a computer paired with the HMD. Products like the Oculus Quest or HTC Cosmos promise a standalone experience that will bring high quality graphics and motion tracking into a portable device, in some cases at a considerably lower price.

Based on the rapid evolution of VR technologies, we considered that implementing compatibility between various HMDs will allow the game to reach a more substantial amount of players, strengthening the VR community and creating a healthier playerbase.

4.2 FUTURE WORK

We feel that we lack the resources to complete the production of the game as a premium first-person shooter / simulator game. We are, however, eager to publish our work and are contemplating turning MechVR into a short demo or even a puzzle game where the player would have to fix the mech to escape it. The resulting demo/game would be published for free in Steam Store for maximum exposure.
As we set out to answer the question, “How to develop an action game that would best translate first-person shooter game mechanics into a virtual reality environment?”, we have documented the production of the prototype game “MechVR” and the key design decisions for game mechanics, controls and mech structure.

The project was started in late 2016, when VR gaming seemed to be on the verge of a commercial breakthrough. Our goal was to produce a complete commercial game, but we ended up shelving the project after 16 months of development. This was mostly due to: 1) lack of players to support a multi-player VR online game, which was our original goal and 2) lack of resources to develop an interesting single-player game, which was our backup plan.

Despite having failed to produce a finished game, we feel that we succeeded in adapting FPS-like gameplay to VR in a novel and unique way. We also believe MechVR's gameplay is especially well-suited for a multi-player online game, in which players fight each other in a last-man-standing scenario, a genre often referred to as “Battle Royale”. We think our work could be used as a foundation for such games, if the market for them would establish itself in the future.
References


Junnila, M. (2007). Genret ja niiden rikkominen pelisuunnittelussa. Available at: https://aaltodoc.aalto.fi/handle/123456789/12484

Typefaces:
Gap Sans by Étienne Ozeray & Alexandre Liziard
   (a fork of Sani Trixie Sans typeface by GrandChaos9000)
IA Writer Quattro by Information Architects Inc. (iA)
Seravek by Eric Olson (Process Type Foundry)

Graphic Design:
Akbar Khatir

Printer:
Picaset Oy