UNIVERSITY OF TARTU

Institute of Computer Science

Software Engineering Curriculum

Kaarel Rüüsak

Minesweeper With Friends – An Immersive Multiplayer Puzzle Game

Master's Thesis (30 ECTS)

Supervisor(s): Mark Muhhin

Tartu 2023

Minesweeper With Friends – An Immersive Multiplayer Puzzle Game

Abstract:

This thesis describes the development and design of an immersive puzzle game that can be played either cooperatively or competitively. As there are few high-quality versions of *Minesweeper* that feature cooperative or competitive gameplay in an immersive 3D environment, the game created throughout this thesis will help to fill that void and provide players with a new immersive and intuitive gameplay experience. The game was made by a single developer within two development iterations, each involving a design, implementation, and testing phase. In the final testing phase, it is discovered that the game achieves a significant degree of immersion, but there are many dimensions of immersion for which the game falls short. This data is then used to propose future improvements for the game's development.

Keywords:

Game, video game, game development, puzzle game, Minesweeper, Unity, playtesting

CERCS: P170 Computer science, numerical analysis, systems, control

Minesweeper With Friends – kaasahaarav mõistatusmäng mitme mängijaga

Lühikokkuvõte:

See lõputöö kirjeldab kaasahaarava mõistatusmängu loomist, mida saavad mängida mitu inimest kas üksteise vastu või ühise eesmärgiga. Arvutimängude turul on vähe kõrge kvaliteediga versioone mängust *Minesweeper*, mis sisaldavad kaasahaaravat 3D keskkonda. See lõputöö aitab täita seda auku turul, luues mängu mis pakub mängijatele kaasahaarava ning intuitiivse kogemuse. Mäng on loodud ühe arendaja poolt kahe iteratsiooni käigus. Igas iteratsioonis disainiti uued mängu omadused, mille põhjal mäng loodi ning seda uuendati. Mõlema iteratsiooni lõpus mängu testiti, et parandada võimalikke probleeme ning leida viise mängu edasi arendamiseks.

Võtmesõnad:

Mäng, arvutimäng, videomäng, mõistatusmäng, Minesweeper, Unity, mängu testimine

CERCS: P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria)

Table of Contents

Introduction 5

1 Competitor Analysis 6

1.1 Microsoft Minesweeper 7

1.2 Bunny Minesweeper 8

1.3 Minesweeper Multiplayer 9

2 First Iteration 11

2.1 Game Design 11

2.1.1 Dimensionality 11

2.1.2 The Minefield 12

2.1.3 The Tiles 13

3.1.4 The Money 14

2.1.5 The Camera 15

2.1.6 The Player Characters 16

2.1.7 The Game World 17

2.2 Implementation 18

2.2.1 Used Tools 18

2.2.2 User Interface 19

2.2.3 Minefield 20

2.2.4 Character Animations 20

2.3 Testing 21

2.3.1 MängudeÖÖ 2021 22

2.3.2 Computer Graphics Project Expo 22

2.3.3 Comic Con Baltics 2022 23

2.3.4 Results 24

3 Second iteration 26

3.1 Game Design 26

3.1.1 Health and Money 26

3.1.2 Survival Mode 27

3.1.3 Lobby Settings 28

3.1.4 Visualizing Destroyed Mines 29

3.1.5 Tutorial 29

3.2 Implementation 30

3.2.1 Codebase Restructuring 30

3.2.2 Survival Mode 31

3.2.3 Character Dismemberment 32

3.3 Online Survey 33

3.3.1 Survey Methodology 33

3.3.2 Survey Results 34

3.4 Second Iteration Testing 35

3.4.1 Testing Methodology 35

3.4.2 Usability Testing Results 36

3.4.3 Feedback Form 39

4 Future Development 42

Conclusions 43

References 45

Appendix 48

I. Glossary 48

II. Accompanying Files 49

III. Source Code 50

IV. Room Setup for Each CGP Expo Event 51

V. License 52

Introduction

Video game development is a growing industry, estimated to increase from 155.89 million dollars in 2020 to 268.11 million by 2025 [1]. Entering such a growing industry is incredibly valuable, but there are over 10,000 games released on Steam every day [2], which makes it difficult for new products to stand out. In order to create a successful video game, independent game developers usually need to innovate and create new products that fill a distinct niche separate from not only mainstream games but also their competitors [3].

Such a gap in the industry has been identified: few developers are making multiplayer puzzle games inspired by *Minesweeper* and fewer still are making immersive 3D implementations intended to be proper experiences instead of simple experiments. The developers who focus on this field rarely seem to innovate with the formula and their projects tend to have critical flaws. Some of these kinds of projects are further analysed in detail in the next chapter, Competitor Analysis.

The goal of this thesis is to fill this gap in the industry by creating a uniquely immersive and intuitive multiplayer game inspired by *Minesweeper* called *Minesweeper With Friends* (*MWF*). The purpose of this game is to garner interest from the general public and build a recognizable brand for the developer. To achieve these goals and ensure that the game is a suitable product for the market, similar competing products are analysed in Chapter 1, Competitor Analysis.

The main competitive advantage of *Minesweeper With Friends* is providing the players with an immersive experience while maintaining the intuitive gameplay elements of the original *Minesweeper*. For the purposes of developing video games, immersion is generally defined as the degree to which players are involved with the game [4]. The effectiveness and appeal of these features are tested several times at different events, altering the game with each development iteration. The first and second iterations of designing, developing, and testing the game are covered thoroughly in Chapter 2 and Chapter 3, respectively. Based on the results of the final testing, many future improvements to the game have also been designed, which are further explained in Chapter 4.

1 Competitor Analysis

It is useful for game developers to analyse competitors, as it helps learn important lessons from the efforts of other developers and create a new strategy [5]. In this chapter, several other implementations of *Minesweeper* are analysed and compared to *Minesweeper With Friends* (*MWF*) in order to find inspiration and to determine whether *MWF* is providing unique value to the players. One game (titled Mine Sweep Battle[1]) was excluded from this section, as it was released after the first development iteration of *MWF*.

As this thesis is intended to maximise the number of players, an estimated number of owners will be calculated using the “Boxleiter” method for competitors whenever it is applicable. This method consists of multiplying the publicly available number of reviews on the digital game distribution platform Steam[2] with an estimate called a “Boxleiter number” [6]. This number was calculated as the mean of the average (63) and median (58) reviews per sale for premium games, compiled by Simon Carless [7]. The resulting ratio is an estimated 60.5 owners of any Steam game for each review. By multiplying the Boxleiter number by the number of reviews for each game, it is possible to estimate the average number of sales even for games that are too obscure to track with other tools such as SteamSpy[3].

However, for this Boxleiter number, it is important to note that the middle 80% of the samples fluctuate between 25 and 100 sales per review [7]. This is a margin of error of about ±37.5 sales/review, which is more than half the size of the proposed Boxleiter number. This implies that the results are still likely to be inaccurate, which is why they will not be entirely relied upon. Instead, the resulting estimated sales numbers will only be used to support theories and lend them some extra credibility. Now that suitable metrics have been established, it is possible to analyse games similar to *Minesweeper With Friends*.

[1] https://thegdwc.com/games/630554d5-715f-4d3e-a601-e7e318e434fc

[2] https://store.steampowered.com/

[3] https://steamspy.com/

1.1 Microsoft Minesweeper

Microsoft Minesweeper is a game that was originally released in 1990 in a collection called the “Windows Entertainment Pack”, after which it was a standard feature in the “Windows” operating system [8]. In Windows 8 and later, the game has no longer been included with the Windows operating system, but the modern version[4] is still being updated with extra game modes, features, and more.

This section analyses the version of *Microsoft Minesweeper* that was included with the “Windows XP” operating system, as it was a direct inspiration for *Minesweeper With Friends*. Additionally, this version of Minesweeper is very similar to several free adaptations of the game such as *Minesweeper X*[5] and *Minesweeper Online*[6].

Fundamentally, *Microsoft Minesweeper* is a logic-based puzzle game about revealing squares in the grid called the “minefield”. At the beginning of the game, a random pattern of mines is hidden in the minefield. Every square in the grid is displayed as a blank, unrevealed space to the player. If the player clicks on an unrevealed square, the number of bombs adjacent to that square is displayed instead (see Figure 1). If they click on a bomb, they immediately lose the game. The player can win the game by revealing every square of the minefield that does not contain a mine.

Additionally, the game has several features that assist in solving the puzzle or enhance the experience. The player can right-click any empty tile to “flag” it, adding a visual flag symbol on the tile and preventing the player from accidentally clicking on the tile. This feature empowers players to mark known mine locations to prevent accidents and record their progress toward solving the puzzle. Furthermore, the number of remaining unmarked mines is displayed in the top-left corner of the application, calculated by subtracting the total number of placed flags from the total number of mines in the game.

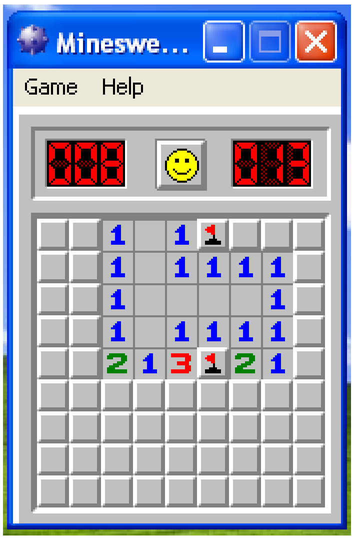


Figure 1. Screenshot of *Microsoft Minesweeper*

[4] https://www.microsoft.com/en-us/p/microsoft-minesweeper/9wzdncrfhwcn

[5] http://www.curtisbright.com/msx/

[6] https://minesweeper.online/

The game features a timer in the top-right corner, counting upwards from zero seconds starting at the moment the first tile was revealed by the player. When the player has revealed all tiles that do not contain mines, if their score was the fastest time for that difficulty recorded on the computer, they were given the opportunity to save their name into a scoreboard. This scoreboard only displays the fastest scores for each difficulty.

1.2 Bunny Minesweeper

Bunny Minesweeper is a fully 3D implementation of *Minesweeper* (see Figure 2) that includes cooperative gameplay. The controls are similar to *Microsoft Minesweeper*, where left- or right-clicking while standing on a tile will either reveal it or flag it as a bomb. The objective for all players is to reveal the entire minefield, after which the players will automatically get cosmetic rewards based on how complex the minefield was, and the final score (based on the time taken to solve the minefield) is saved to a scoreboard.

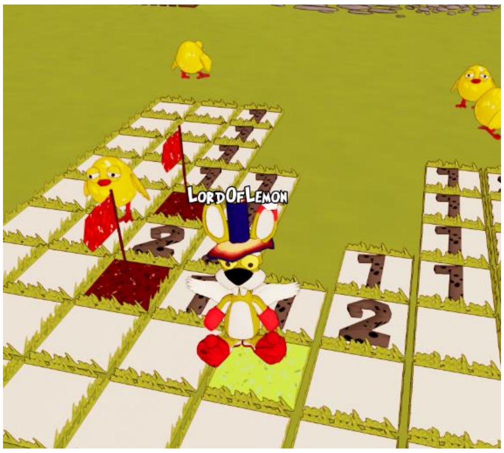


Figure 2. Screenshot of *Bunny Minesweeper*

The game concept is very similar to the classic game mode in *Minesweeper With Friends* and 90% of the reviews for the game are positive[7]. However, the game seems to have fallen short in terms of marketing, as there are only a total of an estimated 3085.5 sales for the game, despite its positive reception. Another sign of its unpopularity is the price, which is just 2.39 € by default, and it has been further discounted to merely 0,47 €. Based on Video Game Insights[8], this price is only about 6% of the current average price for a paid indie game, which is 8.37 €. Such a low price could be an indication that the developer is not confident in their product or its current sales numbers.

Although most of the reviews for the game are positive, several of the negative reviews for *Bunny Minesweeper* lament its graphics and controls. Namely, to reveal tiles, the player character needs to touch them. As these characters move slowly compared to the mouse-controlled system in *Microsoft Minesweeper*, experienced players may be unaccustomed to

[7] https://store.steampowered.com/app/900790/Bunny_Minesweeper/

[8] https://vginsights.com/steam-analytics

the pace of the gameplay. Additionally, this change adds an element of unpredictability to each player's score, as the players may be forced to move long distances repeatedly due to the layout of a minefield.

Bunny Minesweeper features an alternative game mode called "survival". In this game mode, when the players finish solving a minefield, the floor is destroyed, and a new minefield is generated below them to be solved. The game also features a third game mode, titled "crazy", but this is just a more difficult version of the normal game, with huge versions of the minefield (i.e. 50 by 50). In other respects, the extra game modes are identical to the original game, although their scores are tracked on separate scoreboards.

1.3 Minesweeper Multiplayer

Minesweeper Multiplayer[9] is a free game developed by Thomas Heine Rasmussen that can be played in a web browser. It is an interesting take on the *Minesweeper* formula because it not only features competitive multiplayer gameplay, but in this case, players are rewarded for revealing the bombs (displayed as flags) instead of being punished.

The objective of the game is to get more points than the opponent, and the gameplay is completely turn-based. This means that while one player is choosing which tile to reveal, other players cannot do anything. If a player reveals a flag, they receive one point and their turn continues. However, if an empty tile is revealed, that player's turn ends and play continues to the next player.

Additionally, each player starts the game with a single bomb, which they can use whenever it is their turn, and they have fewer points than their opponents. This bomb will instantly reveal a five-by-five area of tiles and end that player's turn. Any score for revealing the flags is awarded to the player who used the bomb.

Despite being innovative and easy to learn, there are quite a few problems with the design and implementation of this game. The most noticeable problem is the lack of momentum in the game itself. Most other *Minesweeper* implementations[10] make the first click on any minefield safe. This has the added benefit of speeding up the early game, quickly reaching an interesting point where players are forced to use their deductive skills to figure out the

[9] https://minesweeper-multiplayer.dk/about

[10] https://minesweepergame.com/strategy/first-click.php

best tiles to reveal. Because *Minesweeper Multiplayer* does not use this approach, the first few clicks can simply reveal a single number and no surrounding squares. In that case, it may take a long time for players to reach a point where interesting gameplay emerges (see Figure 3 below).

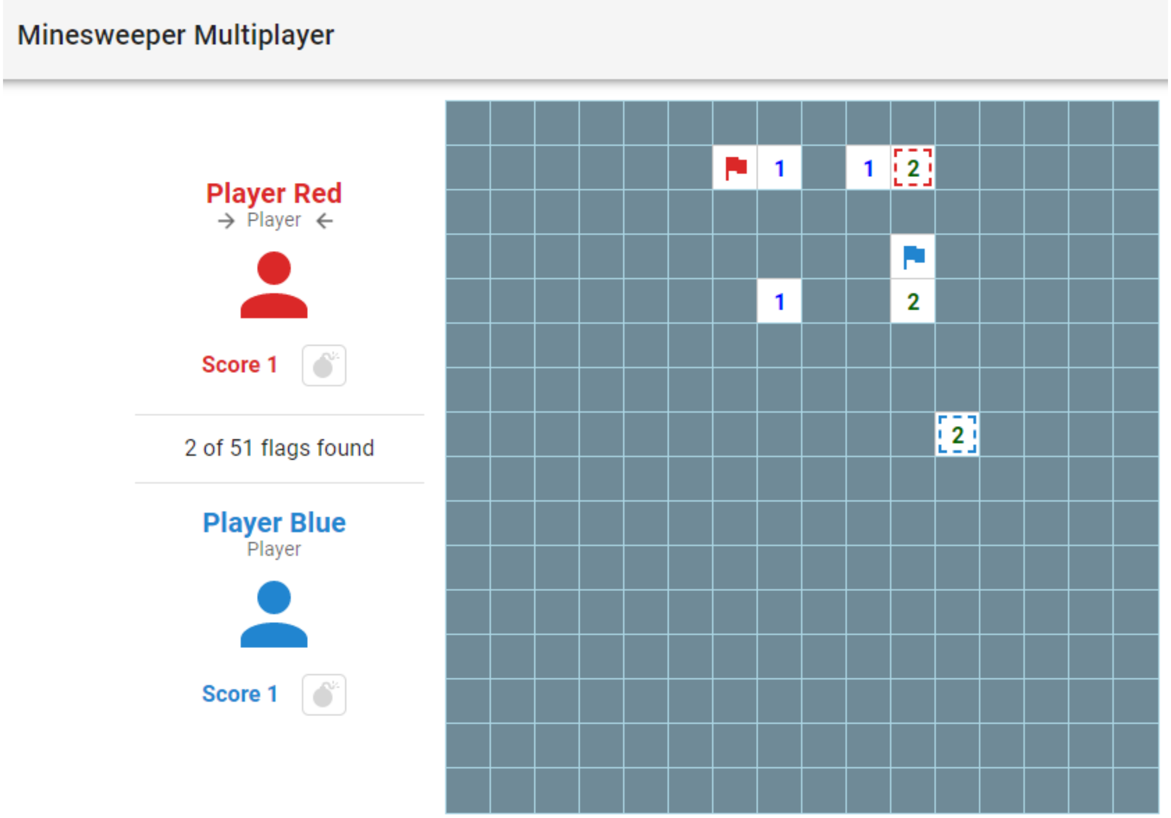


Figure 3. Screenshot of *Minesweeper Multiplayer*

Based on the insights gained from reviewing the competitors and their performance, a new version of *Minesweeper* was created, which offers unique value to players and avoids the pitfalls of the other games. The development of this game is discussed further in the following chapter.

2 First Iteration

In order to ensure the game is suitable for the market, the development of *Minesweeper With Friends* follows an iterative strategy. This means that the development process goes through several cycles of design, implementation, and testing, with the goal of improving the game with each cycle (called iteration). This chapter covers the first development iteration for the game *Minesweeper With Friends*, whereas the second iteration is discussed in Chapter 3. The game's initial mechanics and various design elements are described in Subchapter 2.1. Its technical implementation and the various technologies used for development are explained in Subchapter 2.2. At the end of this development iteration, the game was also extensively tested at several different events, which is covered in Subchapter 2.3.

2.1 Game Design

MWF is a puzzle game designed for two to four players, which is intended to stand out from its competitors by providing the players with a heightened sense of immersion while maintaining the intuitive gameplay of classic *Minesweeper*. Although the game's design was planned out and documented before development, it was rapidly altered as the game was developed and ideas were either discarded or expanded upon. This chapter explains the design of *MWF* from its inception until the end of the first iteration.

The experience of immersion has been positively linked with game appeal [9] and it is an important factor in game enjoyment [4], which implies that an increased degree of immersion provides significant value for players. Therefore, this thesis aims to maximise the degree of immersion whenever it is applicable. One way in which greater immersion can be achieved is by using a three-dimensional environment.

2.1.1 Dimensionality

Three-dimensional games offer a much greater sense of being inside a place than two-dimensional spaces [10]. Additionally, many of the competing multiplayer *Minesweeper* games analysed in the previous chapter are two-dimensional, so adding a third dimension would help visually differentiate *MWF* from its competitors. For these reasons, *MWF* was developed as a 3D shooter, instead of a typical 2D puzzle game.

Furthermore, introducing a third dimension to the game provides more space for implementing visual effects and other gameplay elements. One such example is the usage of large particle effects whenever a player makes progress by revealing tiles, which would usually obstruct the vision of the puzzle in a two-dimensional space. However, in *MWF*, this problem is lessened by scattering the particles in a sparse pattern above the puzzle. Due to their low density and long distance from the puzzle elements, player vision is rarely blocked by the particles (see Figure 4). Furthermore, on the rare occasion that a player's view of the puzzle is directly obstructed by the particles, the player would only need to adjust their camera angle in order to see behind the effects.

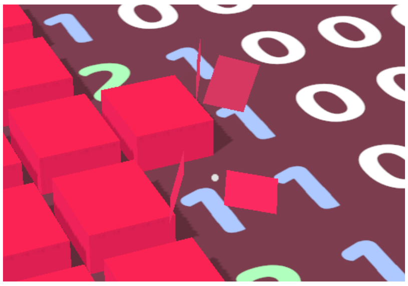
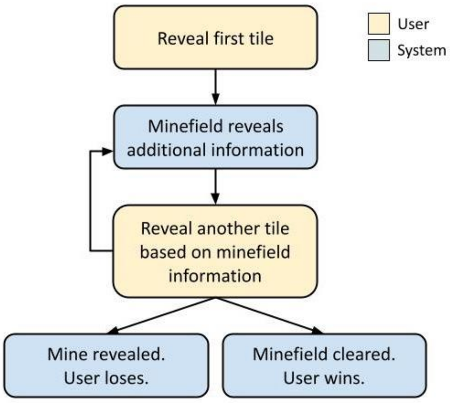


Figure 4. Particle effects in *MWF*

2.1.2 The Minefield

In a classic game of *Microsoft Minesweeper*, the minefield is a central design element that players spend most of their time interacting with. Whenever players reveal a tile on the minefield, they are directly rewarded with more information about nearby mines, allowing players to continue solving the puzzle. This creates a gameplay loop (see Figure 5), which directs the entire flow of gameplay.

In *Minesweeper With Friends*, the minefield serves the same purpose and therefore shares many characteristics with the minefield in *Microsoft Minesweeper*, such as the grid-based design and the aforementioned gameplay loop. Furthermore, both games present a large amount of numerical information to the players, which is much easier for players to visually parse when the information is colour-coded. For this reason, each revealed number on the minefield has been assigned a colour depending on its value.



Reveal first tile
User
System
Minefield reveals additional information
Reveal another tile based on minefield information
Mine revealed. User loses.
Minefield cleared. User wins.

Figure 5. Gameplay loop of *Minesweeper* games

Another feature that is common to both *Microsoft Minesweeper* and *MWF* is the way the mines are distributed within the minefield. If every mine was placed in a random location, it would be possible for a player to lose when revealing the first tile in the game. For this reason, *Microsoft Minesweeper* ensures that the first revealed tile is guaranteed to be empty. *MWF* takes this a step further by ensuring that the first tile and all the tiles connected to it are empty. This means that the uninteresting beginning of most *Minesweeper* implementations is skipped, which would normally involve players clicking on random squares and restarting the game until they find a "lucky" pattern of mines.

2.1.3 The Tiles

According to the American video game designer Jesse Schell [11], the skills a player exercises are very important for determining the nature of a player's experience. In *MWF*, the most interesting core skill required from the players is the mental challenge of solving the minefield by quickly calculating the risk and reward for clicking on unrevealed tiles. In order to allow players to focus more on this mental challenge, the physical challenge of aiming and clicking on each tile was simplified as much as possible.

For this reason, each tile on the minefield is as large as reasonably possible in all three dimensions, leading to a box-shaped design. These boxes are still small enough to leave noticeable gaps between each tile, which is helpful for counting the number of potential bombs when players are trying to solve the puzzle. Additionally, these boxes are just low enough to the ground that from most camera angles, they aren't obstructing the visibility of the numbers, which are crucial for solving the puzzle.

Highlighting the unsolved part of the puzzle has other benefits as well. As Scott Rogers [12] has mentioned in his book, "Level Up! The Guide to Great Video Game Design", it is a good practice to direct players toward their path of progression using geometry and lighting. For this reason, the unsolved tiles on the minefield have been tinted bright red. This naturally directs players toward the part of the puzzle they can solve, making the gameplay loop more intuitive to follow.

Furthermore, using bright red tiles means the players will gradually remove the most glaring colours from the scene throughout the course of gameplay. As they are doing so, the colour scheme of the game will gradually shift from one that highlights the unsolved tiles into one that is darker and more soothing. As Rogers [12] has mentioned, using different colour schemes is an effective way to make players feel the changes in the world. Therefore, the usage of bright red colours in the design of unsolved tiles is likely to have the added benefit of enhancing the player's sense of progression.

This same principle is also followed in the visualization of flagged tiles. Players in most implementations of *Minesweeper* are able to highlight unrevealed tiles which they believe to contain mines by right-clicking them, which is known as flagging mines. In order to allow players to focus on logical thinking instead of memorization, *MWF* also implements this feature. Furthermore, in order to make the colour palette of a solved minefield appear more soothing, flagged mines in *MWF* are tinted in a deeper, darker shade of red (see Figure 6).

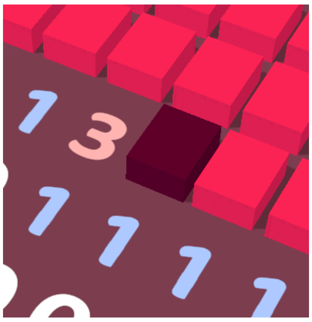


Figure 6. Flagged tile

3.1.4 The Money

An important aspect of game design is figuring out how to balance the game so that unskilled players don’t become frustrated when learning how to play [11]. This is especially difficult in a puzzle game such as *Microsoft Minesweeper*, where even a single mistake could mean instant failure. For this reason, instead of instantly killing any player who has made a mistake, *MWF* uses a different system.

Whenever players in *MWF* make a mistake by clicking on a mine, they are deducted one bundle of money, which is an in-game currency. Players start with three bundles and when they have no money left, they enter debt, which means they are no longer able to reveal tiles. Once a player has reached this point, their only means of progression are to restart the game or convince other players to give them money.

Players can earn more money by solving the puzzle. Whenever a tile is revealed, the revealing player earns a number of money bills depending on the number that was revealed. Therefore, sections of the puzzle with a higher density of mines will grant more money, creating a situation of increased risk and reward.

Once any player has gathered a sufficient number of money bills, they are automatically converted into a money bundle, which can be used to survive mistakes or given to other players. This system ensures that new players have several opportunities to experiment and make mistakes without being forced to restart the game. However, in order to further ensure that playing the game and earning money remains an enjoyable experience, it is necessary to use a suitable camera perspective for the game.

2.1.5 The Camera

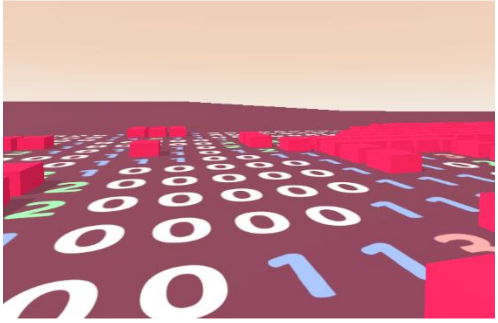


Figure 7. Approximation of a first-person view

During the course of the game, a camera is the lens through which the player perceives the world. As discussed by Rogers [12] in his book, nothing will stop players from playing faster than a poor camera. This is why it is crucial to use a camera that is convenient to control and complements the gameplay, instead of hindering the player.

Initially, the implementation of a first-person camera was considered, which would place the camera in the same position as the player's eyes. Naturally, as this view is akin to what people experience in real life, it can provide some players with an enhanced sense of immersion [13]. However, after some consideration, it was discovered that a first-person view would restrict players from seeing any part of the minefield distant from the player (see Figure 7). This made it more difficult to focus on solving the puzzles and reduced the speed of gameplay, which is why a different type of camera was used.

In order to show more of the world while still using characters to represent the players, a third-person camera was used in *MWF*. However, most third-person camera implementations still don't show much of the environment, as they keep the camera close to the player and prevent players from looking directly downwards. This practice enhances immersion and is usually effective in third-person shooters and action games because in such games players are typically moving and shooting horizontally.

However, players in *MWF* need to spend a lot of time looking down and making logical deductions based on the information on the ground. For this reason, the camera's pivot point was moved high above the player. Additionally, the camera was moved far away from its pivot point, further increasing the maximum distance between the minefield and the player's view. In practice, this means that players are always looking at the world from an angle far above their character (see Figure 8), which provides players with a better view of the puzzle throughout the game. Furthermore, the camera's pivot point was moved slightly in front of the character instead of directly above them. This allows players to look directly downward without being obstructed by their character.

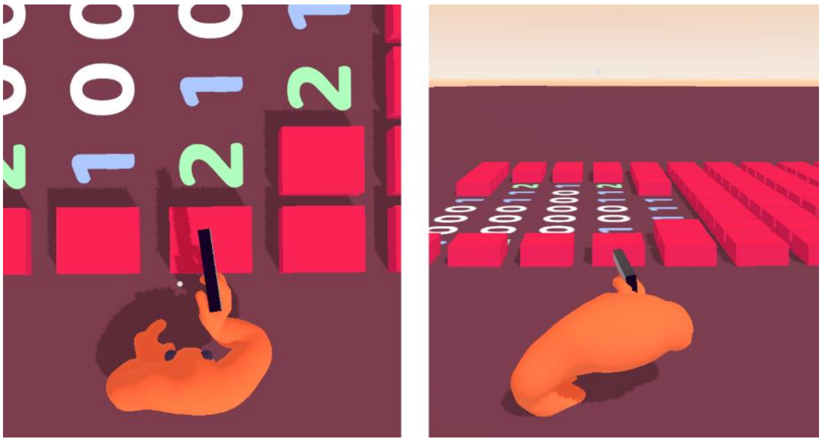


Figure 8. Camera in *Minesweeper With Friends*

2.1.6 The Player Characters

A player character, also known as an avatar is a representation of the player within the game world. In games that feature a third-person camera, designing an entertaining character is crucial, as players will spend a lot of time inspecting their character [10]. For this reason, the character in *MWF* was designed to be non–humanoid and weird. Therefore, the base of the player was created, which consists of two muscular arms, ending at three fingers. This unusual design also helps the game aesthetically stand out from its competitors.

This character was also improved to make it more attractive to human sensibilities. It has been proven that certain features such as large eyes and a small mouth or nose increase the appeal of a character's face [14]. For this reason, player characters in *MWF* have large, spherical eyes and a small mouth. The resulting character has distinctly humanoid features in a strange, but personable combination (see Figure 9). The general style of this character is cartoonish and abstract, which allows players to better project themselves into the character [13]. Once a player character was created, it was necessary to build an environment that matches the character's visual style.

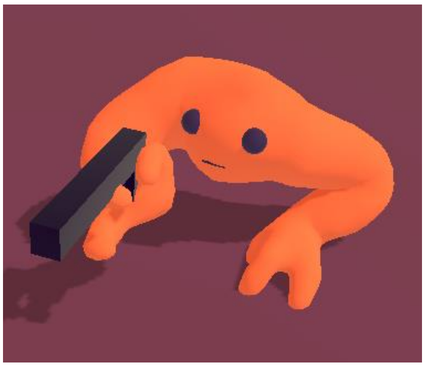


Figure 9. Player character

2.1.7 The Game World

In order to better immerse players in the game while matching the visual style of the characters, a simple and abstract scene was created for the events of *MWF* to take place. The visual style of this scene was inspired by games like Human: Fall Flat[11] and Gang Beasts[12]. This means that the game's world features simple, abstract textures and models, reducing the memory requirements for running the game. Additionally, the game does not require the usage of post-processing effects such as depth of field or motion blur, further improving the performance of the game on low-end machines.

In order to keep players focused on the minefield and avoid any obstructions to the camera, *MWF* is set in a large, open area. However, because the game was intended to be as immersive as possible, a plausible location with some ambient motion was used, which makes the world appear to be more alive. For these reasons, *MWF* is set on a large empty island in the middle of a moving ocean (see Figure 10 below). The movement of the ocean is slow enough that it doesn't needlessly distract from the puzzle-solving gameplay, but it is still fast enough to be noticeable.

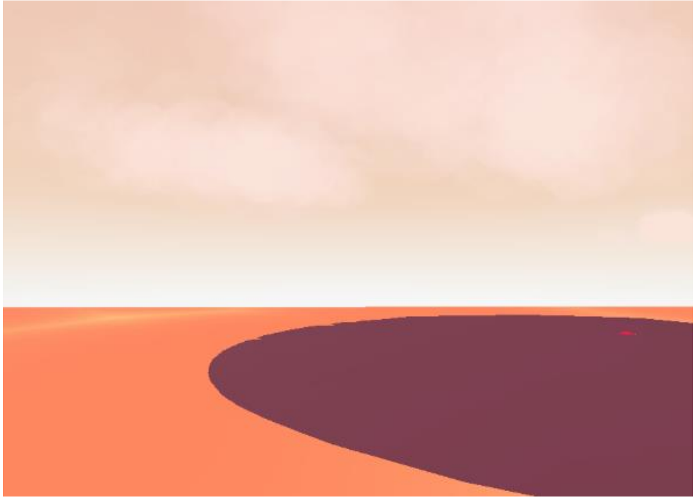


Figure 10. The world of *MWF*

[11] https://store.steampowered.com/app/477160/Human_Fall_Flat/

[12] https://store.steampowered.com/app/285900/Gang_Beasts/

2.2 Implementation

The first playable version of *Minesweeper With Friends* (*MWF*) was designed and developed within a span of a few months as a proof of concept. At this stage of development, the goal of this project was simply to verify whether the game concept has significant appeal and potential. Despite starting on a small scale, the project gradually grew into a fully-fledged game because more time and effort was devoted to developing the idea. To support both fast prototyping and further development without wasting development time, it was important to use tools that were suitable for efficiently developing the game.

2.2.1 Used Tools

Several game engines, such as Unreal Engine, Godot, and Unity were considered for the development of the game. Unity[13] was picked as the game engine for developing *MWF* because it has a wide range of community-made assets and extensions, which are excellent for complementing the needs of a developer who is working alone. Additionally, this decision was supported by the developer's previous experience with this game engine from courses at the University of Tartu courses and personal projects. The scripts for the game were written in C# using the Visual Studio Code[14] editor.

There are many packages for quickly implementing multiplayer in Unity. At the start of *MWF*'s development cycle, there were two options built into the game engine for doing so. One of these solutions was UNet, which had already been deprecated for several years [15]. The newer alternative, Unity Multiplayer Networking, was still in development and unstable. Because of this, an open-source third-party extension called Mirror Networking[15] was used instead.

The 3D graphics and modelling software called Blender[16] was used for creating the various 3D models necessary for development. Additionally, Gimp[17] was used to create rasterized graphics or textures for the game. In situations where vector graphics were more appropriate,

[13] https://unity.com/

[14] https://code.visualstudio.com/

[15] https://mirror-networking.com/

[16] https://www.blender.org/

[17] https://www.gimp.org/

Inkscape[18] was used. Blender, Gimp, and Inkscape are all open-source software. This means that they are open for modification and developed by the community, often without being paid for their work. As such, the author of this thesis extends their gratitude to the many open-source developers who helped improve these projects and make them what they are today. Using this software, many different 3D and 2D assets were created, one of which was the user interface.

2.2.2 User Interface

To provide an interface for the game, Unity's default interface system was used. This choice was made in the initial prototyping stages, as the system was readily available, free, and provided the necessary functionality for creating a minimum viable product (MVP). The implementation of this interface relied heavily on built-in Unity interface elements. Whenever possible, interface elements were set up to interact directly with each other, without reliance on any custom scripts. However, as development progressed, it was apparent that these tools were insufficient for creating all the animations necessary for creating an appealing game.

In his book "Game Feel", indie game developer Steve Swink [16] general principles for creating games that feel good to play, some of which are instantaneous response, predictability and novelty. According to these principles, in order to create a game that exhibits good game feel, it is important to create an instantaneous, noticeable response to the input. This response should be predictable to the player but have enough variety to remain novel.

To put these principles into practice, all the buttons in *MWF* are rotated temporarily whenever the player's mouse cursor starts hovering over them. This effect is pretty simple and predictable to the player, but the angle of rotation is random in order to create some novelty in the animation. Additionally, whenever a new menu is opened, the new buttons are scaled upwards from zero and rotated in an analogous way. With these additions, the interface is more dynamic and provides the players with a greater sense of game feel, enhancing the experience of using it.

In order to implement these features, Unity's interface had to be extended with custom scripts. A script called `UIAnimations` was created for this purpose, which is designed to

[18] https://inkscape.org/

be general enough to work in any Unity project. The script contains a set of animations, and it can be attached to interface buttons in the game.

With this script, it is possible to use the Unity editor to define a series of custom animation curves. These curves are used in the script's various animation functions, which rotate or scale the object attached to the script. These functions can optionally be triggered whenever the object is enabled. Alternatively, they can be called directly through Unity's event triggers or other scripts. This functionality and most other animations featured in *MWF* were created using a free open-source animation engine called DOTween[19].

2.2.3 Minefield

As the central design element, the minefield is one of the most complex parts of the game. The script responsible for initially generating the minefield is called `GridSystem`, as it generates a grid of empty tiles at the start of the game. The size of this minefield is determined by the parameters assigned to the script in the Unity editor. Initially, this script does not contain any information about mine positions.

Whenever the first tile is revealed, the layout of the mines is calculated for the entire minefield based on the position of the revealed square. These mines are randomly distributed among every tile not adjacent to the initially revealed square. In order to prevent clients from using external tools to cheat, the mine layout is only stored on the server and any operation that requires access to it is likewise performed on the server.

Whenever a new tile is revealed, a series of operations are performed on the server to check whether the game is over. Because many independent systems such as the scoreboard and user interface need to be informed of the game ending, the *observer pattern* was used for its implementation. The programming language C# features "events," a ready-made solution for using this programming pattern, which was used to implement the *observer pattern* in *MWF*.

2.2.4 Character Animations

Usually, video game characters are animated as a ready-made sequence of motions that are replicated by the character models. However, this method has significant drawbacks, as it is difficult to smoothly combine the animations in a way that appears natural in every

[19] http://dotween.demigiant.com/

circumstance. Because of this, the arm motions for characters in *MWF* are instead moved exclusively using scripts. This practice is inspired by the independently developed game Rain World[20], which uses a similar method to achieve dynamic animations for dozens of non-humanoid characters.

To avoid creating scripts for dozens of character bones, Unity's default implementation of inverse kinematics (IK) was used. IK is a method of animation where the desired position of the end effector (in this case the hand) is given to the animation and the rest of the motion is automatically calculated [17]. In practice, this means the positions of character hands and faces were determined by custom scripts and the rest was automatically done by Unity.

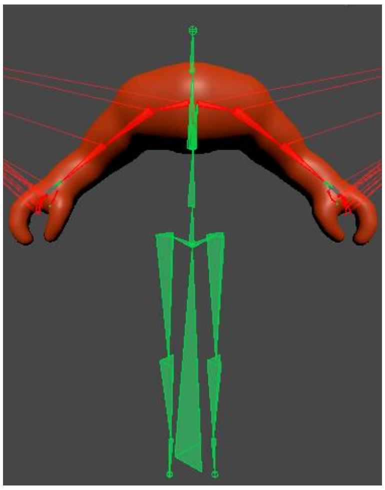


Figure 11. Player character model rig

However, Unity only supports IK for humanoid characters, which was a problem. The characters in *MWF* have no head or legs, therefore they are not humanoid. This issue was overcome by building a humanoid bone structure around the existing character (see Figure 11). The bones for "missing" limbs were not assigned to the model and therefore didn't affect the character at all, but Unity still recognized the resulting model as humanoid. This enabled the usage of Inverse Kinematics to automatically animate character elbows.

2.3 Testing

At the end of the first iteration, it was necessary to test the game and gather feedback to validate the idea and further improve the game. For this reason, the game was shown off at several events and each event helped uncover many small problems, which were fixed after the event. Therefore, the game presented at each event was in a slightly different state, but the game's general design principles and the testing process remained the same.

The game was also altered slightly to be suitable for testing at such events. Every version of the game used during these tests has a hotkey that restarts the game without disconnecting any of the players. This allows anyone present to quickly restart the game when a different person wants to try out the game.

[20] https://www.gdcvault.com/play/1023475/Animation-Bootcamp-Rainworld-Animation

Furthermore, in order to foster competition and make the events more exciting, a simple scoreboard was added to the game. This scoreboard displays the names of the top players who solved the minefield the fastest, along with the time it took them to finish the game. The first of these playtests was conducted at MängudeÖÖ 2021.

2.3.1 MängudeÖÖ 2021

This event was held on the evening of September 25, 2021. At the time, *MWF* was just a playable prototype, and the goal of this test was to prove whether the game concept was fun and engaging. As such, the players were provided instructions about how to play the game on paper (see Appendix II), which allowed new players to learn the game despite not having previous experience with other *Minesweeper* games.

The testing area occupied the wall of a corridor, which led to the rest of the event. Tables were placed along this wall to hold the necessary equipment. *MWF* was playable cooperatively on two laptops placed on this table. Additionally, one of these laptops was connected to a large widescreen monitor along the wall to display a live feed of the gameplay to people passing by. Next to the laptops were small pages of paper with controls and instructions for playing the game written on them.

The test was very successful, with many players enjoying the game and some even returning several times to improve their scores. At the event, the game was played by over 30 people and most testers enjoyed the game. However, the concept of using money as health seemed difficult for some players to grasp. This test also highlighted several performance bugs, which were fixed later in development.

2.3.2 Computer Graphics Project Expo

After receiving feedback at MängudeÖÖ, several bugs and performance problems for the game had been fixed. Additionally, the game was visually improved within the scope of the University of Tartu course "Computer Graphics Project" (CGP), increasing the overall visual appeal of the game. The course was taken twice, during both the spring and fall semesters of 2022. At the end of each semester, students had to participate in an expo to display the results of their projects, which is where these playtests were organised. The second CGP expo occurred early in the second development iteration, but as its results were very similar to the first expo, they are both covered in this chapter.

During the first CGP Expo, six desktop computers were set up to play the game, whereas the second expo only had room for two computers. In addition to *MWF*, both events had many other games set up to play Minesweeper. The room configurations for each event can be found in Appendix IV. During both events, players were provided paper instructions identical to the ones used at MängudeÖÖ 2021. Additionally, a small leaflet was placed on a stand next to the computers, displaying a description of the project and its goals.

Compared to other events, attendance for these expos was low: about 10 to 20 people per event. However, most of the attendees were students or employees of the University of Tartu and they often had a background in science, so they were eager to give detailed feedback. In addition to uncovering many bugs, the detailed feedback from these playtests helped inspire many future features for the game, such as the Survival Mode. Furthermore, the first CGP Expo was a partial inspiration for developers of another multiplayer version of *Minesweeper*[21], which was later nominated for multiple awards at the Game Development World Championship[22].

2.3.3 Comic Con Baltics 2022

This expo was held on May 20-22 in 2022 at the Lithuanian Exhibition and Congress Centre. Throughout the event, *MWF* was playable on two computers placed on a standing desk (see Figure 12). At the time, the game was an improved version of the prototype, with enhanced graphics, animations, and some bug fixes. Players were provided the same paper instructions used during previous events.

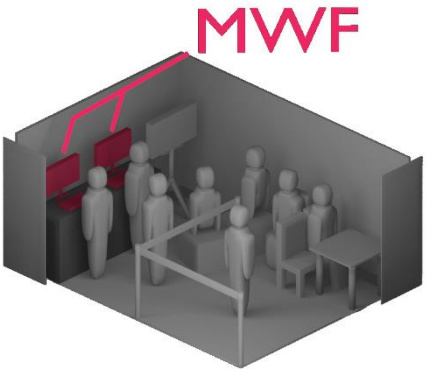


Figure 12. Comic Con Baltics 2022 room setup

During the event, the game was tested extensively over the course of three days by a total of over 50 people. Several bugs and problems were found within the game, which were fixed later. Most notable examples include:

1. The computers overheating due to resolution scaling problems;
2. Custom display settings sometimes reset when players restart the game;

[21] https://mihkelroomet.itch.io/mine-sweep-battle

[22] https://thegdwc.com/games/630554d5-715f-4d3e-a601-e7e318e434fc

3. Tiles sometimes disappear from the minefield when players restart the game.

When the players were asked what they think would improve the game, two experienced players suggested adding chording to the game. Also, many players had suggested a competitive game mode, where players are able to shoot each other. This is likely due to the somewhat confusing nature of the harmless pistols every player carried, but it still sounded like an intriguing idea. This was one of the inspirations for the game's Survival Mode, which is discussed further in the next iteration.

2.3.4 Results

Despite the different locations, the testing process at each event was very similar and there were at least 100 total participants. Over the course of these tests, certain notable patterns of behaviour emerged among the players most interested in the game. Two distinct types of invested players were observed: nostalgic speedrunners and curious explorers.

Most of the nostalgic speedrunners (average age 18-24) were used to playing the original game and wanted to compete for achieving the fastest time, also known as speedrunning. This score was their main motivation and they often returned to the testing booth several times to beat their old scores, or the scores of other players with a similar goal. At each test, between one and four nostalgic speedrunners were observed. Sometimes, when these players happened to meet at the same time, they cooperated in order to create a new high-score which would become higher than their individual scores. Such scores were usually at the top of the leaderboards until the end of the event, or until a different combination of speedrunners beat them.

The curious explorers (average age 12-24) were a type of player that was more interested in learning and experiencing the game. In every test, at least five of these types of players were observed. These players were very curious and offered many creative suggestions for game improvements, despite usually having little to no previous experience with *Minesweeper*. The most common suggested feature brought up by more than five independent players from this category was the ability to fight or compete with their friends directly in some way. After offering their suggestions, these types of players usually played for a length of time longer than the average player, but shorter than that of most speedrunners.

Such behaviour may suggest that the curious explorers were interested in the game, but it wasn't offering them an engaging enough experience to keep playing. To remedy this and to further engage players who were looking for new experiences, an alternative game mode with a more directly competitive focus called “Survival Mode” was created, which is described in more detail in Chapter 3.1.

However, both types of players encountered a problem with mines that had already exploded. To solve the minefield, players would often have to count the number of mines that surround a specific square. While doing so, players would often disregard revealed mines, because only a two-dimensional sprite was left behind. This would lead to several mistakes, which confused and frustrated players. This problem was later remedied by creating a three-dimensional model for a tile that has exploded, which is described further in Chapter 3.1.

Another notable discovery was that players often encountered problems when they tried to learn the game without reading the instructions. It was common for players to reveal tiles randomly and fail to understand that they were losing money with each mistake. Once they ran out of money, this would cause confusion and it would sometimes even lead to these types of players giving up on the game. In order to tackle this problem, the entire system of money was overhauled with a new health system in the next iteration of development.

3 Second iteration

During the second iteration of the game's development, both the game's design and technical aspects were improved. New features, such as a new game mode and a health system were designed, which is described further in Chapter 3.1. The technical implementation of these features and some problems encountered during that process are discussed in Chapter 3.2. In order to figure out the game's suitability for the market, an online survey was conducted and analysed in Chapter 3.3. Once the improved version of the game was playable, detailed user testing was performed, which is discussed further in Chapter 3.4.

3.1 Game Design

In the tests performed in Chapter 2.3, several important shortcomings of *MWF*'s game design were highlighted. In order to further improve the game, many features of *MWF* were added or redesigned based on this feedback. This chapter describes the many design decisions made during the game's second development iteration, the first of which was replacing the existing system for money.

3.1.1 Health and Money

In the tests performed in the previous iteration, players were often unable to figure out why and how they were losing money when they made mistakes. The connection between the money and its diegetic usage was tenuous and difficult to understand even for experienced fans of other PC games. To make this connection more intuitive for players to grasp, the game's mechanics were brought closer to what is expected from games in the same genre.

MWF can be categorized as a third-person shooter and a puzzle game. Neither of these types of games typically make players lose money when they make mistakes. Instead, in most third-person shooters such as *Warframe*[23] and *Evil Dead: The Game*[24], players lose health whenever they make mistakes. In order to make *MWF* easier to understand, a similar health system was added to the game, which replaces some of the functionality from the previously implemented money system.

[23] https://store.steampowered.com/app/230410/Warframe/

[24] https://store.steampowered.com/app/1493750/Evil_Dead_The_Game/

Like in the first iteration of the game, players in *MWF* will earn money by solving the puzzle. However, when players accidentally reveal a mine, they now lose one of three lives. If a player has lost all their lives, they are unable to reveal tiles until they buy more. Lives can be purchased in a new store menu for one bundle of money per life (see Figure 13).

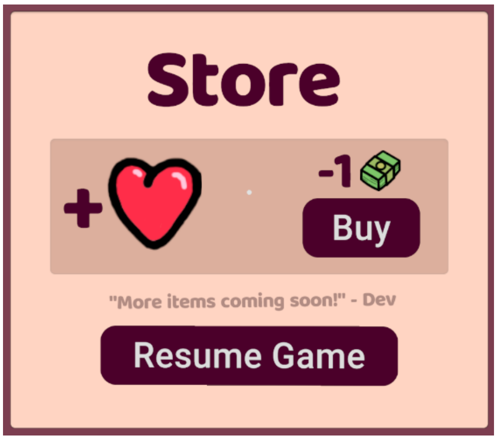


Figure 13. Store menu

To ensure that players intuitively understand and become aware of their character's health, it is also useful to reflect it in their appearance [12]. For this reason, health in *MWF* was also displayed as limbs exploding off the character (see Figure 14). The removed limbs would become skeletal versions instead, with the character becoming completely skeletal when all health is lost. This system also has the benefit of displaying each player's health to other players, allowing them to see who is keeping pace and who may need help.

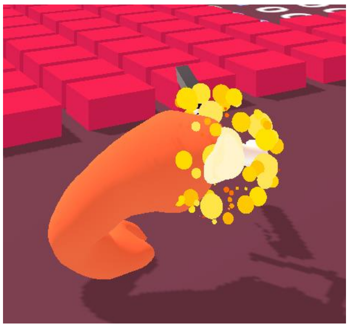


Figure 14. Character's limb exploding

However, this change introduced another type of resource that players would need to keep track of (health), which made the game more complex. To reduce some of that complexity and keep the game as intuitive as possible, leaves of money were removed from the game. Instead, players can now only see the bundles of money they have collected. This also adds an element of surprise to the process of revealing tiles, as players do not know exactly which tile would "contain" the next bundle of money.

3.1.2 Survival Mode

The testing performed at various events during the last development iteration revealed that many players have the desire to play the game competitively. To cater to the needs of these players, a new game mode called Survival Mode was added to *MWF*. This is a new way to play the game, differing from the cooperative game mode developed in the previous iteration, which is now called Classic Mode. When a player starts hosting a new game, they can choose which of these game modes they would like to play.

Survival mode does not feature health or money. Instead, each player has a timer which is counting downward from 30 seconds (see Figure 15). Whenever players reveal a tile, they are rewarded with extra time on this timer. However, when the timer reaches zero, players are unable to reveal any tiles. In this game mode, player limbs are destroyed and regrown depending on how much time they have left. In this way, the timer serves as a replacement for all the functionality of the health system.

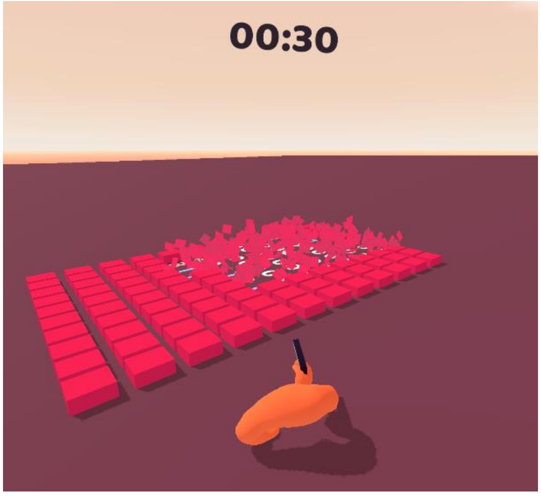
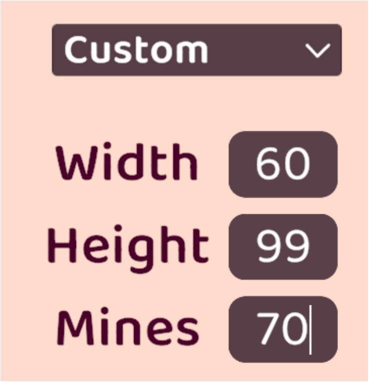


Figure 15. Survival mode timer starting

The game lasts until every player except one is unable to continue solving the puzzle. At that point, the player who can still play is the winner. Currently, the game can also end with everyone winning the game if the minefield is solved before the end of the game. However, in the future, this minefield will be replaced with a new version that periodically replaces solved tiles with unsolved tiles, gradually increasing the game's difficulty until someone wins.

3.1.3 Lobby Settings

In games like Minesweeper, it is difficult to adjust the game's challenge to be appropriate for its audience. As the testing in the previous chapter has shown, the potential audience for the game can vary significantly in terms of their experience and skill, which is important to consider when designing the game's difficulty [10]. After all, a game that provides an appropriate challenge for one player may be stressful for another.



Custom

Width 60

Height 99

Mines 70

Figure 16. Lobby settings

To alleviate this problem, *MWF* allows players to choose difficulty levels, which is a well-tested and reliable way to balance a game [11]. This allows players to decide upon the appropriate level of challenge by themselves and pick the appropriate difficulty level (easy, medium, or hard). Because of this, the scoreboard was also separated into three different sections depending on the difficulty.

To provide players with even more freedom, they can also create custom games, which allow them to choose the minefield's length, width, and mine density (see Figure 16). However, because the difficulty can vary so significantly in this case, the scoreboard becomes practically useless for the custom difficulty levels. As such, the custom difficulty is not useful for players whose main goal is to improve their score.

3.1.4 Visualizing Destroyed Mines

During the tests conducted in the previous development iteration, it was discovered that players would often overlook revealed mines. This was a significant problem, as the process of solving the minefield would often require players to count the number of mines in an area. If players overlook a destroyed mine, they may make an error in counting the number of mines, which would often lead to a mistake in solving the puzzle, causing frustration for the players.

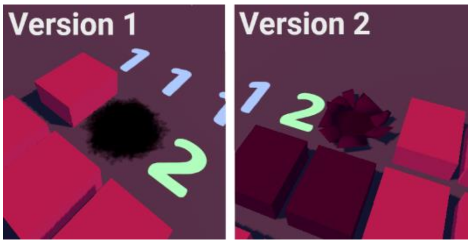


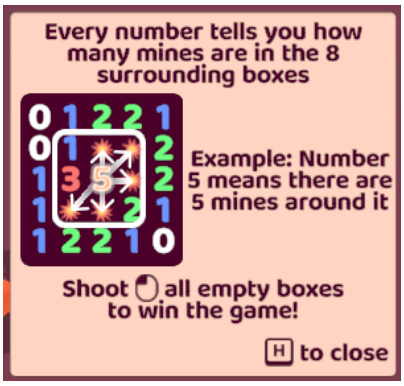
Figure 17. Versions of the explosion craters

To remedy this problem, destroyed mines needed to be visualized in a more noticeable way. Initially, an experimental two-dimensional crater sprite was used. However, because all tiles were three-dimensional, this would still lead to many players overlooking the two-dimensional sprites when counting the number of mines. For these reasons, a second, three-dimensional version of the crater was created (see Figure 17). The color of this model is the same as the flagged tiles, which is likely to be an intuitive connection between craters and flagged tiles.

3.1.5 Tutorial

During the previous iteration's testing phase, the game was very difficult for some players to understand without the use of paper instructions. Because most players will not have access to these paper instructions, it is necessary to provide sufficient hints to the players using the game itself. For this reason, some simple and non-intrusive tutorial hints were added to the game.

The tutorial includes simple prompts for using the various controls necessary for playing the game. Whenever the player presses the prompted keys, the tutorial moves on to the next instruction. Additionally, a new menu that is accessible by pressing “H” was added to provide new players with information about how *Minesweeper* works (see Figure 18). Combined, these tutorials provide players with all the information provided by the paper instructions used for testing the game’s first iteration.



Every number tells you how many mines are in the 8 surrounding boxes

Example: Number 5 means there are 5 mines around it

Shoot all empty boxes to win the game!

H to close

Figure 18. Tutorial menu

3.2 Implementation

Throughout the second development iteration, a lot of new functionality was designed and developed. This subchapter covers the technical implementation of the many features that were designed in the previous section. In order to support all these features, the game’s scripts were restructured to improve the overall structure of the codebase, which is explained further in Subchapter 3.2.1. Additionally, a new competitive game mode was created, which is explained in Chapter 3.2.2. Furthermore, the development of a health system that involves the player avatar’s limbs being destroyed is covered in detail in Subchapter 3.2.3. Additionally, many of the small technical problems found in the previous iteration’s testing phase were resolved throughout the development of the second iteration.

3.2.1 Codebase Restructuring

At the start of the game’s development, *MWF* was a quick prototype created as a simple proof of concept. As such, much of the codebase was unstructured and difficult to navigate. To implement the various features described in the design chapter, some improvements were made to the codebase in order to make it follow a more logical structure and become easier to manage.

The first of these improvements was the usage of assembly definitions[25], which are assets that can be used within the structure of a Unity project to organize scripts into assemblies. By doing so, scripts can be detached from other scripts, allowing each assembly to be

[25] https://docs.unity3d.com/Manual/ScriptCompilationAssemblyDefinitionFiles.html

compiled separately. This could reduce compilation times, as only a portion of the codebase would need to be refactored when small changes are made to scripts. Additionally, the implementation of assembly definitions served as an inspiration to decouple some scripts from each-other scripts, resulting in code that was more modular.

Once this structure was established, the code was also organized into namespaces[26]. Namespaces allow code to be organized so that only relevant code is accessible from any part of the codebase. Whenever a script requires the usage of scripts from another namespace, they need to explicitly declare this with a "using" statement at the start of the script. Thus, well-structured namespaces have the additional benefit of making the code easier to intuitively understand, as any script's dependencies could be determined by examining its "using" statements. To make this type of system easier to use, the namespaces were aligned with the project's folder structure, and they are kept automatically synchronised using a free Unity Asset Store tool called SmartNS[27].

3.2.2 Survival Mode

In addition to the aforementioned codebase changes, many more changes had to be made in order to support an additional game mode. Many scripts related to the game modes, such as the `GameScenePlayer` and `PlayerDataManager`, were changed into abstract or inherited classes. This approach was used to provide separate functionality dependent on the selected game mode. Whenever applicable, any scripts that were specifically used for a single game mode were named with an appropriate suffix. For example, the inherited version of a script called `GameScenePlayer` created specifically to provide functionality for the Survival game mode would become `GameScenePlayer_Survival`.

As many of these changes required significant code refactoring or restructuring, this process took a lot of development time. As the code was being refactored for these purposes, several small code quality improvements were also made. For the most part, this involved changing function or variable names to be more specific, removing unused code, and generally improving readability.

[26] https://docs.unity3d.com/Manual/Namespaces.html

[27] https://assetstore.unity.com/packages/tools/utilities/smartns-109633

3.2.3 Character Dismemberment

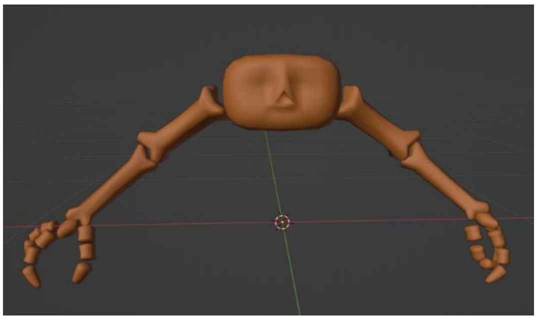


Figure 19. Skeletal character model

Another feature, which took significant effort to implement was the visualization of player health as limbs falling off the character. For the player character to keep interacting with the world as they are losing limbs, an alternative skeletal version of the player model was created using Blender (see Figure 19). Once implemented, the skeletal version was intended to always be beneath the player character's skin. This way, as the skin of a specific limb is removed, the skeletal version of that limb would be revealed underneath.

However, there were some problems with implementing this system. Although the skeletal version of the player's model was created to be covered by skin from all angles, this did not apply to the model when it was posed. In most cases, bending any of the character's limbs would cause parts of the skeleton model to poke out from underneath the skin (see Figure 20).

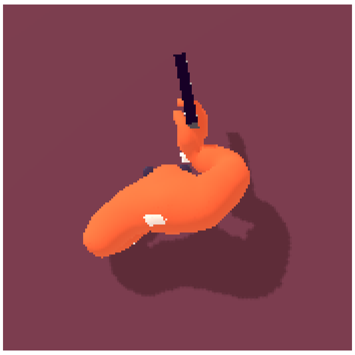


Figure 20. Character bones protruding through skin

To resolve this problem, two new scripts were created: `PlayerLimbsDisplay` and `Limb`. Both are responsible for hiding and revealing different parts of the skeleton as the player loses health. Additionally, these scripts remove the player's skin from different limbs when necessary. This is achieved either by scaling the relevant bones to zero or by hiding the models altogether.

3.3 Online Survey

Based on the testing performed in Chapter 2.3, two distinct types of players were identified: nostalgic speedrunners and curious explorers. This subchapter aims to further validate these observations and find additional information about *MWF*'s potential future players. For these purposes, an online survey was created with Google Forms[28] and sent out to a community of *Minesweeper* fans. The creation of this survey is explained further in Chapter 3.3.1. The results are analysed and discussed in Chapter 3.3.2.

3.3.1 Survey Methodology

It has been discovered that retro game features are likely to be a positive signal to consumers [18], creating a more favourable and nostalgic attitude toward the game. Based on this information and the observations described in Chapter 3.3, players who are nostalgic toward the older versions of *Minesweeper* are likely to try out new games that provide a similar experience, such as *MWF*. Because of this, the survey contains four questions about previous respondent experience with *Minesweeper*.

To figure out the demographics of the game's potential players, the survey also includes two questions about respondent demographics and four questions about respondent gaming habits. Furthermore, to better direct future marketing activities, many additional questions were included about the usage of social media platforms, news websites, and gaming habits. The final survey has a total of 19 questions and takes less than ten minutes to fill out.

To gather useful feedback, this survey had to be sent to a large audience of players invested in who may be interested in the game. Thus, the link to the survey was shared with Reddit's dedicated *Minesweeper* community[29], which has over 12000 members. After it was published, the post was upvoted by thirteen people and it received one comment, which led to a significant number of participants filling out the survey.

[28] https://forms.google.com/

[29] https://www.reddit.com/r/Minesweeper/

3.3.2 Survey Results

A total of 38 respondents answered the survey over the course of eight days. This number surpasses that of most previously conducted testing sessions. Additionally, the long survey helped discover many details about respondents, that the previously conducted tests did not uncover. Furthermore, because most of the respondents are dedicated fans of *Minesweeper*, they are likely to be a more accurate representation of *MWF*'s target audience than the attendants of an unrelated physical event.

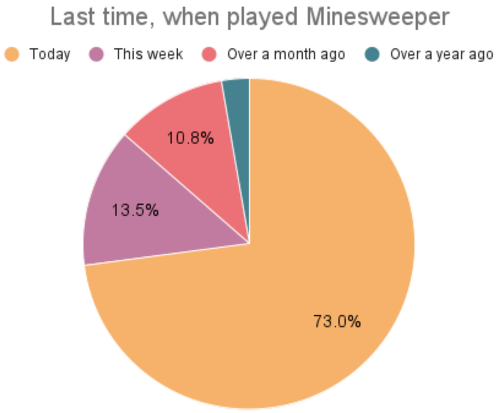


Figure 21. Last time participants played *Minesweeper*

Over 87% of respondents had played some version of *Minesweeper* within a week of answering the questionnaire (see Figure 21) and all 38 respondents agreed that they have previously enjoyed playing other versions of *Minesweeper*. This result was likely because the survey was conducted among a community of *Minesweeper* fans. This means that survey participants line up with the image of "nostalgic speedrunners" that formed during the testing performed in Chapter 2.3.

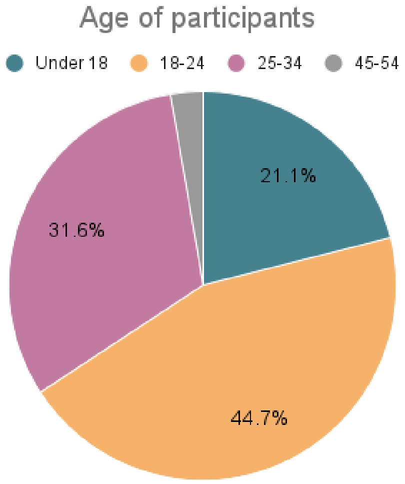


Figure 22. Participant age groups

The most common age group for respondents was 18-24, followed by 25-34 (see Figure 22). These two age groups combined constitute over 76% of the respondents. Based on these results, *MWF*'s potential target audience is people ages 18 to 34 who have previously played other versions of *Minesweeper* and are looking for something new.

Over a third of the respondents play games on their laptops. Furthermore, a survey by Statista [19] showed that over 70% of adults in the United Kingdom have a laptop. Therefore, it is likely that a large portion of potential *MWF* players would prefer to play the game using their laptops.

However, most laptops are generally slower than desktops[30]. Thus, the ability to run the game on low-end machines like laptops would give *MWF* a significant competitive advantage compared to other games with more detailed graphics and worse performance. This also means that optimization should be an important factor in *MWF*'s future development. To further improve the course of the game's development, testing of the game's second development iteration was performed.

3.4 Second Iteration Testing

Throughout *MWF*'s second development iteration, several important features were reworked or added to the game. To determine potential usability problems for the resulting game, physical user testing of the game was conducted. Subchapter 3.4.1 discusses the physical testing methodology, whereas Subchapter 3.4.2 analyses the results of these tests. To further validate these findings and to determine the degree to which players were immersed in the game, a feedback form was created. The results of this feedback form are further discussed in Subchapter 3.4.3.

3.4.1 Testing Methodology

One of the most useful ways of usability testing is observation, which requires only five users to provide meaningful data [20]. Because of this, observation was used as the means to determine whether *MWF* is free of usability problems and intuitive for players to understand. Thus, participants were invited to a physical location where they could play the game and solve any problems they encountered on their own. This process was observed, as any significant setbacks encountered by players were written down for further analysis.

Additionally, if players consented, then playtests were recorded for later review. For these purposes, the open-source recording and broadcasting software OBS[31] was used. A total of eleven participants were recorded playing the game. However, on several occasions, there were either technical problems with the recordings, or the participants asked for their tests not to be recorded.

[30] https://www.technowize.com/lets-end-the-foolish-debate-of-laptop-vs-desktop/

[31] https://obsproject.com/

Most of the testing was conducted in the University of Tartu Delta Centre, within the Computer Graphics and Virtual Reality Lab[32]. However, as the lab was used for an event during a portion of the testing, two testing sessions were carried out outside of this Lab. Regardless of location, the setup included two to four computers with the game already open and connected through Steam to the same lobby for the Classic game mode.

Players were able to adjust lobby settings and start the game whenever they were ready for the test. After participants had finished playing at least one Classic game, they were asked to stop, as the computers were reconnected to a new lobby for Survival mode. After winning or losing at least one game of each game mode, participants were asked to fill out a feedback form.

To organize these testing sessions, the scheduling platform Calendly[33] was used, as it provides a flexible interface for selecting an appropriate date and time. Testers were able to pick any available date and time between April 24 and April 30. Because the game is designed for multiplayer gameplay, each testing session involved two to four participants. Any participants who didn't have a friend to bring along were encouraged to join an existing group of less than four people. Invitations to join the testing were sent out to the University of Tartu's computer science student e-mail list and the Estonian Game Development[34] Discord community.

3.4.2 Usability Testing Results

The game was tested from 22.04 to 28.04 in the year 2023. During this period of time, eight testing sessions were conducted, with a total of 19 participants. Considering the testing period lasted only a week, this was a significant turnout of people, providing more than enough information to make meaningful observations about usability.

These testing sessions usually lasted 15 to 25 minutes, but some participants chose to play more than one game of any game mode or to keep playing after filling out the survey. This practice was encouraged, as it provided additional opportunities for observing player behavior and finding usability problems. In total, 32% of players stayed after the initial testing session, usually because they wanted to try to improve their own score on the scoreboard.

[32] https://cgvr.cs.ut.ee/

[33] https://calendly.com/

[34] https://gamedevestonia.ee/

Table 1 below shows the number of participants for each testing session, along with the number of participants who continued playing after the end of each testing session.

Table 1: Distribution of participants in each testing session

Session #	Testing Session	Number of Participants	Number of Participants who Continued Playing
1	24.04.2023 18:00-18:30	4	1
2	24.04.2023 18:30-19:00	2	0
3	25.04.2023 15:00-15:30	2	0
4	27.04.2023 11:00-11:30	3	3
5	27.04.2023 13:00-13:30	3	0
6	29.04.2023 20:00-20:30	3	2
7	29.04.2023 21:30-22:00	2	0

These testing sessions revealed five potential problems with usability. Additionally, one technical problem related to game functionality was uncovered. The first testing session was much more productive than any other session at uncovering these problems. This can be attributed to the fact that only the first user is required for finding out almost a third of all there is to know about the design's usability [21]. Details about each uncovered issue can be found in Table 2 below.

Table 2: Issues discovered throughout the usability testing process

Issue #	Session #	Description	Priority
Usab-1	1	Survival mode still has the store, despite the shop being useless in that game mode. This confused some players.	High
Usab-2	1	It should be possible to return from the game to the lobby in order to change lobby settings	High
Tech-1	1, 2	Sometimes, players are unable to open the store in Classic mode.	Very High

Usab-3	3	The health of other players should be included as part of the user interface, so players could make faster decisions	Medium
Usab-4	1, 3, 4, 5, 6	Some players are unable to intuitively understand Survival mode	High
Usab-5	5	Some players tried to start solving from different corners of the minefield. This is inefficient because only the first shot reveals a large area.	Low
Usab-6	6	The default mouse sensitivity is too low for most players.	Low

The highest priority issue discovered during the testing process was Tech-1. This issue involves players being unable to open the store in Classic mode. This is a significant problem, as the player who encounters this issue will be unable to buy any additional lives, and thus they may be forced to give up on a game early, despite having the resources to continue solving the puzzle. However, the cause of this issue has not been identified, as it is very rare and it is yet to be reliably reproduced.

Another notable issue is Usab-4, that involves the inability of some players to understand Survival mode. It was previously presumed that a single timer ticking down would make players intuitively assume that the timer reaching zero means their own death. However, most players were unable to understand this connection between the timer and their death. Although in sessions 2 and 7, some players understood the concept and explained it to the others usually this mechanic was too confusing. In such cases, the purpose of the timer was explained to each player.

Other high-priority issues are Usab-1 and Usab-2. The first of these issues was caused by an oversight in the design and development of Survival mode, which allows players to open the store even when it is useless to them. Issue Usab-2, however, causes significant downtime for the players. This issue is caused by players needing to reconnect every time they wish to adjust the lobby's settings, which breaks their immersion within the game's world.

3.4.3 Feedback Form

After each test, the participants were requested to fill out a feedback form. This form provided testers with an additional opportunity to anonymously disclose criticism about the game and suggest potential new features. Additionally, to determine the characteristics of this testing group and whether they fit into the game's target audience, it included several questions about participant demographics and gaming habits.

Furthermore, a section of this form was dedicated to determining the degree to which players of *MWF* were immersed in the game, as immersion is one of the game's most important competitive advantages. This section of the form was created based on the seven dimensions of immersion as defined by Hua Qin *et al.* [22]. However, the dimensions of comprehension and familiarity were excluded from this process, as the game currently does not feature a concrete story that players could understand or familiarize themselves with. Additionally, several questions about immersion were added to the form based on the research conducted by Charlene Jennett *et al.* [23].

More than half of the test participants were between the ages of 18 and 24 (see Figure 23). Furthermore, 68% of participants were between the age of 18 and 34, which matches the game's potential target audience as determined by the online survey conducted in 3.3. In addition, 68% of participants play games on their computers once a week. Therefore, it can be assumed that a significant subset of the participants is part of the game's target audience.

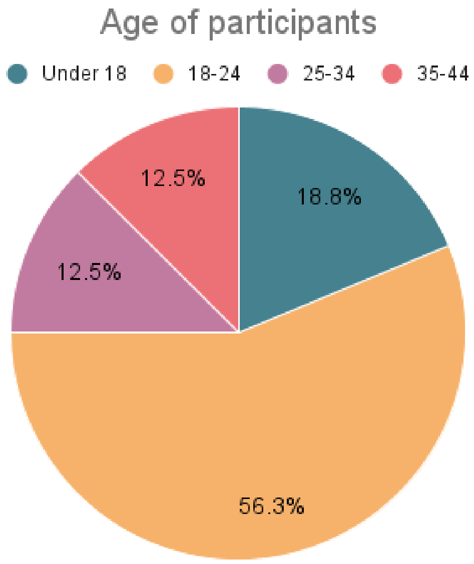


Figure 23. Age of test participants

To the question "how difficult was this game", all the responses were between one and seven on a ten-point scale. Additionally, when asked how well they understood the game, all participants except one responded with values above five. Based on this information, it can be assumed that most participants understood the game and none of them found it notably difficult to play.

Over three-quarters of all participants responded that they would play the game at least once a month in their free time. Furthermore, when asked how likely they are to recommend the game to someone else, three participants (16%) responded with the maximum score on a ten-point scale (see Figure 24). This indicates that many test participants are interested enough in the game to continue playing, and some of them would even go out of their way to promote the game to other players.

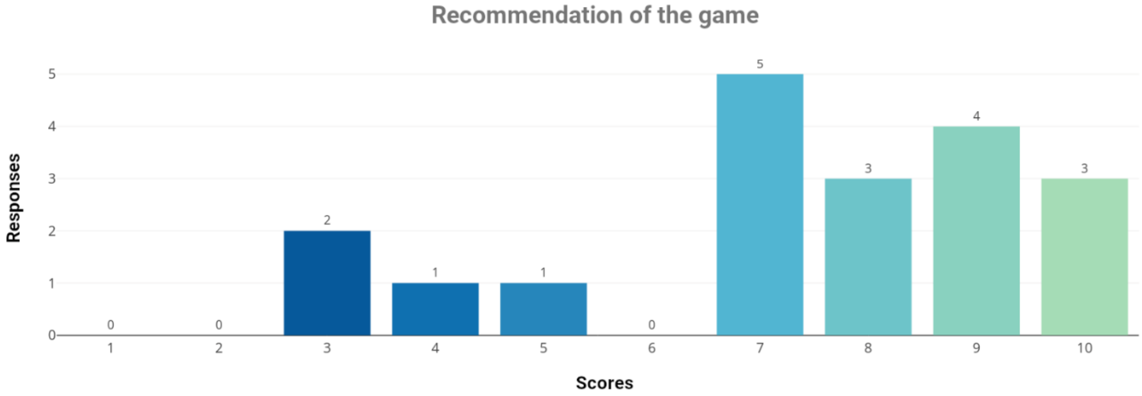


Figure 24. How many participants would recommend the game to their friends

In the section of the feedback form which dealt with measuring the game’s immersion, each question was intended to measure a specific factor of immersion. Many scores of these factors had a distribution that was concentrated at a point above the score of five (see Figure 25 below). The most notable of these are regarding the measurements of player concentration, sensation of character control, and empathy towards the world of the game. Based on these results, *MWF* achieves a high degree of immersion in at least these three out of seven dimensions of immersion.

However, the distribution of some answers to these questions was unexpectedly even. This indicates that the game still has a lot of room for improvement in many dimensions of player immersion. The most notable of them are the following:

- Curiosity about the game and the characters;
- The sense of controlling the game intuitively without being aware of the controls;
- Empathy towards the player avatar.

Additionally, the implementation of a comprehensive story that is familiar to players would likewise improve player immersion, as the related dimensions of immersion had to be excluded from this feedback form.

The final section of this feedback form requests participants to rate the importance of features that may be added to the game in the future. Based on answers to this question and other information gathered throughout this testing process, potential future improvements to *MWF* are discussed in the next chapter.

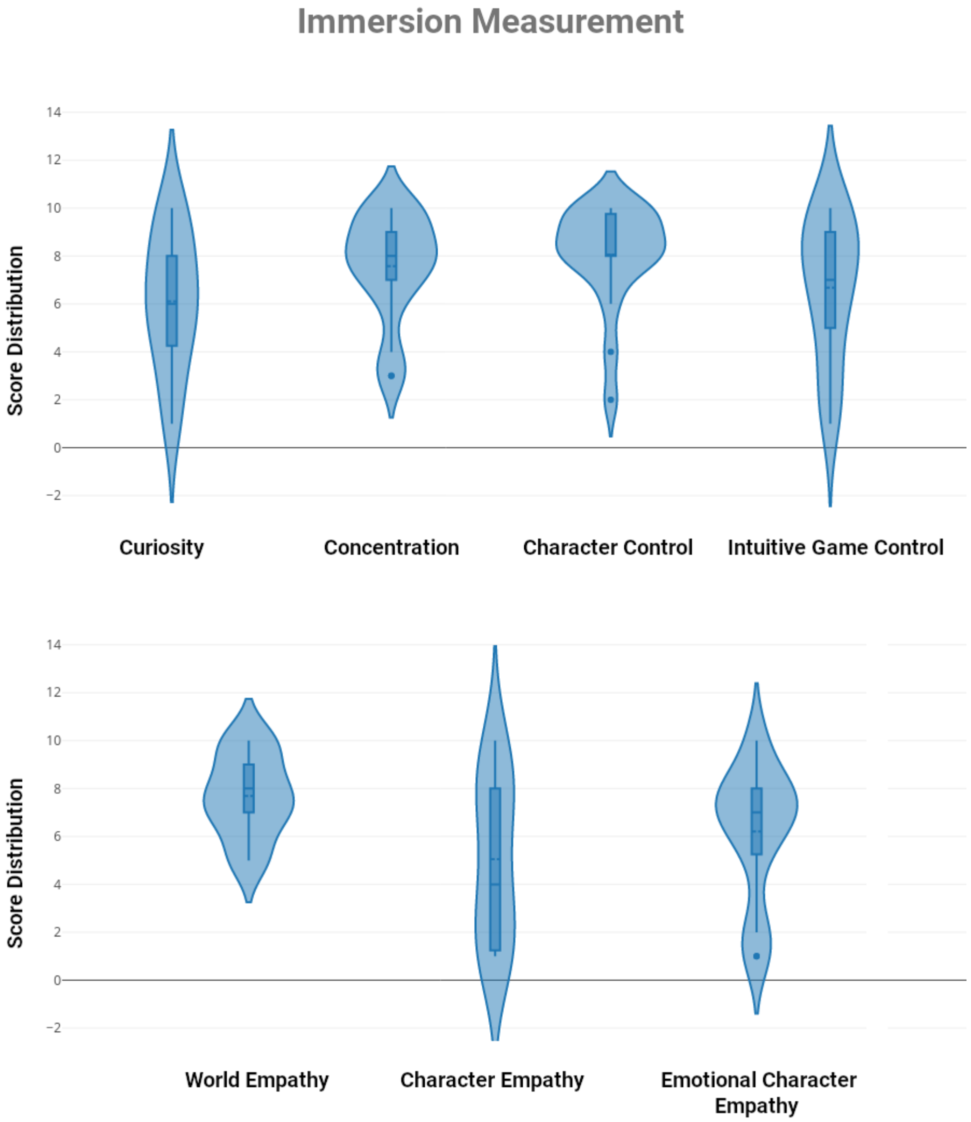


Figure 25. Distribution of answers about the dimensions of player immersion

4 Future Development

During the testing of the second iteration, a lot of new information was uncovered about potential paths of future development for the game. The most urgent aspect of the game's future development is fixing all the very high or high priority issues, followed by most of the medium priority issues. This means that the next phase of development will start with fixing a series of high impact usability problems, along with one technical bug which is currently occurring for an unknown reason.

Soon after resolving these problems, the game will be published on the digital game distribution platform Steam for free. Some work has already been done to integrate the game with Steam's features, such as using the "friends list" feature in Steam to join another player's lobby. However, further development is required to integrate the game's scoreboard with the platform, as well as to create an achievement system to reward players through Steam. In addition, any potential technical issues that emerge from this software integration need to be resolved.

After finishing this integration process, several new features will be added to the game. One of the most requested potential features by players is the implementation of items or powerups into Survival mode, which would allow some players to distract or disrupt other players. This would increase the amount of interaction between different players, providing them with a more dynamic experience with a social focus.

Another highly requested feature was additional options for customizing the player avatar and the environment. These changes would add visual variety to the game, which would make the game seem less repetitive. Additionally, some of these customization options can be linked to player achievements, requiring players to complete an achievement in order to use them. This will give players more incentives to work toward as they play the game, increasing the game's longevity.

Conclusions

Throughout this thesis, a new immersive multiplayer puzzle game called *Minesweeper With Friends* (*MWF*) was created. It was designed to be as immersive and intuitive as possible, in order to provide players with a unique experience distinct from the game's competitors. Development of this game followed two iterations, each of which involved a design, implementation, and testing phase.

Within the first development phase, many features were created for the game in order to improve player experience. Most notably, a system was created that allowed players to earn money throughout gameplay and use it to survive their mistakes. This made the game much more forgiving than most other implementations of the game, usually preventing situations where players would lose the game due to bad luck.

At the end of the first development iteration, the game was playtested at four events, with a total of over 100 test participants. After each event, the game was slightly improved either technically or visually. However, the game's general design principles remained the same throughout this process. These tests were incredibly useful and they served as inspiration for most of the new features created in the game's second development iteration.

During the second development iteration, the system of money was modified and a new health system was added to the game. Furthermore, a store menu and a tutorial were created for the game, along with a new competitive game mode and several other features. As these features were implemented, an online survey was conducted among fans of similar games to determine the characteristics of the game's target audience.

In order to measure the immersion and usability of the game created throughout this thesis, thorough playtesting was conducted at the end of the second development iteration. The tests revealed that although the game manages to achieve a degree of player immersion, there are still many ways in which its immersion could be improved. Additionally, it was discovered that many players enjoyed the game, sometimes to the point of staying after the testing sessions to continue playing. Furthermore, the test uncovered a total of six usability problems and one technical problem, many of which will be fixed in the near future.

After fixing these problems, the future development of the game involves publishing the game on the digital game distribution platform Steam. During this process, the game will be integrated with Steam's systems to better use the tools provided by this platform. Afterward, several new features will be added to the game, most notable of which are usable items or powerups, new customization options, and achievements.

I would like to thank all the testers for trying out the game and providing useful feedback. I thank the University of Tartu's Computer Graphics and Virtual Reality Lab for providing the space and equipment necessary for conducting the second iteration's testing. I would also like to thank Raimond-Hendrik Tunnel for providing useful learning materials about writing the thesis and for inviting me to many events where the game could be tested. Special thanks to supervisor Mark Muhhin for answering many questions and providing useful feedback throughout the writing process.

References

[1] Statista GmbH. (2021, May) Global video game market value from 2020 to 2025. Accessed: Dec. 28, 2022. [Online]. https://www.statista.com/statistics/292056/video-game-market-value-worldwide/

[2] Statista GmbH. (2023, February) Number of games released on Steam worldwide from 2004 to 2022. Accessed: Apr. 7, 2023. [Online]. https://www.statista.com/statistics/552623/number-games-released-steam/

[3] Jesper Juul, *Handmade Pixels: Independent Video Games and the Quest for Authenticity*. Cambridge, England: The MIT Press, 2019, pp. 34-37.

[4] Emily Brown and Paul Cairns, "A grounded investigation of game immersion," in *CHI '04 Extended Abstracts on Human Factors in Computing Systems*, New York, 2004, p. 1298.

[5] Bencin Studios. (2022, December) How to Stand Out - Making a Competitive Analysis in Games. Accessed: Dec. 28, 2022. [Online]. https://www.artstation.com/blogs/bencinstudios/aN0a/how-to-stand-out-making-a-competitive-analysis-in-games

[6] Jake Birkett. (2018, May) Using Steam reviews to estimate sales. Accessed: Dec. 28, 2022. [Online]. https://www.gamedeveloper.com/business/using-steam-reviews-to-estimate-sales

[7] Simon Carless. (2020, August) How that game sold on Steam, using the 'NB number'. Accessed: Dec. 28, 2022. [Online]. https://newsletter.gamediscover.co/p/how-that-game-sold-on-steam-using

[8] Richard Cobbett. (2009, May) The most successful game ever: a history of Minesweeper. Accessed: Jan. 16, 2023. [Online]. https://www.techradar.com/news/gaming/the-most-successful-game-ever-a-history-of-minesweeper-596504

[9] Georgios Christou, "The interplay between immersion and appeal in video games," *Computers in Human Behavior*, vol. 32, no. C, p. 99, March 2014.

[10] Ernest Adams, *Fundamentals of Game Design*, 3rd ed., Karyn Johnson, Ed. United States of America: New Riders, 2014, pp. 140, 62-63, 307.

[11] Jesse Schell, *The Art of Game Design: A Book of Lenses*, 3rd ed. United States of America: CRC Press, 2020, pp. 192, 219.

[12] Scott Rogers, *Level Up! The Guide to Great Video Game Design*, 2nd ed. United States of America: John Wiley and Sons, Ltd, 2014, pp. 240, 84, 131, 102-103.

[13] Katherine Isbister, *How Games Move Us: Emotion by Design*. Cambridge, England: The MIT Press, 2016, pp. 14, 36.

[14] Li Zhu Luo, Hong Li, and Kang Lee, "Are children's faces really more appealing than those of adults? Testing the baby schema hypothesis beyond infancy.," *Journal of experimental child psychology*, vol. 110(1), p. 122, September 2011.

[15] Brandi House. (2018, August) Evolving multiplayer games beyond UNet. Accessed: May. 6, 2023. [Online]. https://blog.unity.com/technology/evolving-multiplayer-games-beyond-unet

[16] Steve Swink, *Game Feel: A Game Designer's Guide to Virtual Sensation*. United States of America: Morgan Kaufmann Publishers, 2009, p. 297.

[17] Rick Parent, *Computer Animation: Algorithms and Techniques*, 3rd ed. United States of America: Morgan Kaufmann Publishers, 2012, p. 192.

[18] Hoon S. Choi, Myung S. Ko, Dawn Medlin, and Charlie Chen, "The effect of intrinsic and extrinsic quality cues of digital video games on sales: An empirical investigation," *Decision Support Systems*, vol. 106, p. 93, February 2018.

[19] Statista GmbH. (2022, August) Share of adults who own a laptop in the United Kingdom (UK) in 2022, by age. Accessed: May. 5, 2023. [Online]. https://www.statista.com/statistics/956332/ownership-of-laptops-uk/

[20] Jakob Nielsen. (2012, January) Usability 101: Introduction to Usability. Accessed: May. 10, 2023. [Online]. https://www.nngroup.com/articles/usability-101-introduction-to-usability/

[21] Jakob Nielsen. (2000, March) Why You Only Need to Test with 5 Users. Accessed: May. 11, 2023. [Online]. https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/

[22] Hua Qin, Pei-Luen Patrick Rau, and Gavriel Salvendy, "Measuring player immersion in the computer game narrative.," *International Journal of Human-Computer Interaction*, vol. 25, pp. 117-118, February 2009.

[23] Charlene Jennett et al., "Measuring and defining the experience of immersion in games," *International Journal of Human-Computer Studies*, vol. 66, no. 9, pp. 657-660, September 2008.

Appendix

I. Glossary

Game mode

An alternative way in which a game can be played. This typically involves separate rules and mechanics which differentiate one game mode from another. This can even extend to the point at which different game modes may belong to different genres altogether.

Multiplayer

A type of game that involves more than one player.

Cooperative Game

A type of game that involves more than one player working together.

Competitive Game

A type of game that involves more than one player working against each other.

Diegetic

Existing within the context of the fictional world in which the game's events are taking place.

Achievement

In the context of video games, a set of arbitrary challenges created by the developer to be met by the player[35].

User Interface

The space where humans and machines interact with one another[36].

[35] https://en.wikipedia.org/wiki/Achievement_(video_games)

[36] https://en.wikipedia.org/wiki/User_interface

II. Accompanying Files

- */paper_instructions* – Folder containing pictures of the paper instructions given to players for testing conducted in the first development iteration
- *gameplay.mp4* – Short video demonstrating the game's features
- *MWF_build_steam.zip* – The zip archive containing the build of the game tested at the end of the second development iteration. It is possible to use this build to connect with other players using the Steam friends list
- *online_survey.pdf* – The online survey used for the market research conducted at the end of the second development iteration
- *online_survey_responses.csv* – The data gathered from responses to the online survey for market research
- *testing_form.pdf* – The feedback form used for the testing conducted at the end of the second development iteration
- *testing_form_responses.csv* – The data gathered from responses for the physical testing feedback form

III. Source Code

Available via request on https://bitbucket.org/KaarelRyysak/minesweeper-with-guns/src/.

Please send an email to kaarelkaks@ryysak.com to request access.

IV. Room Setup for Each CGP Expo Event

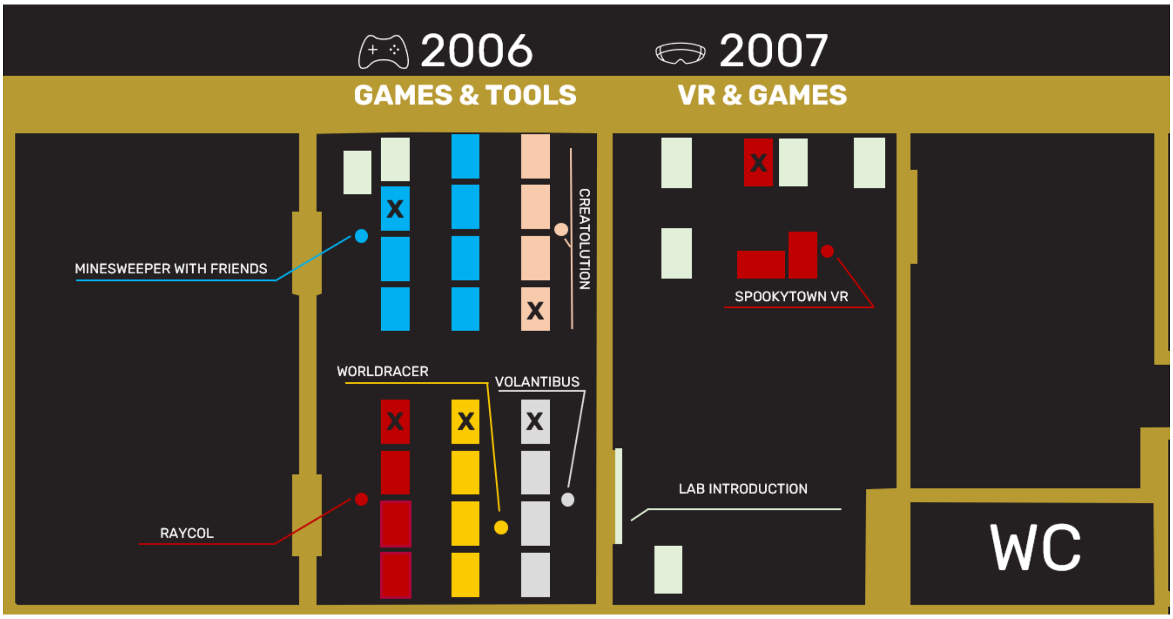


Figure 26. Room setup for the first CGP Expo event.

Author: Daniel Nael.

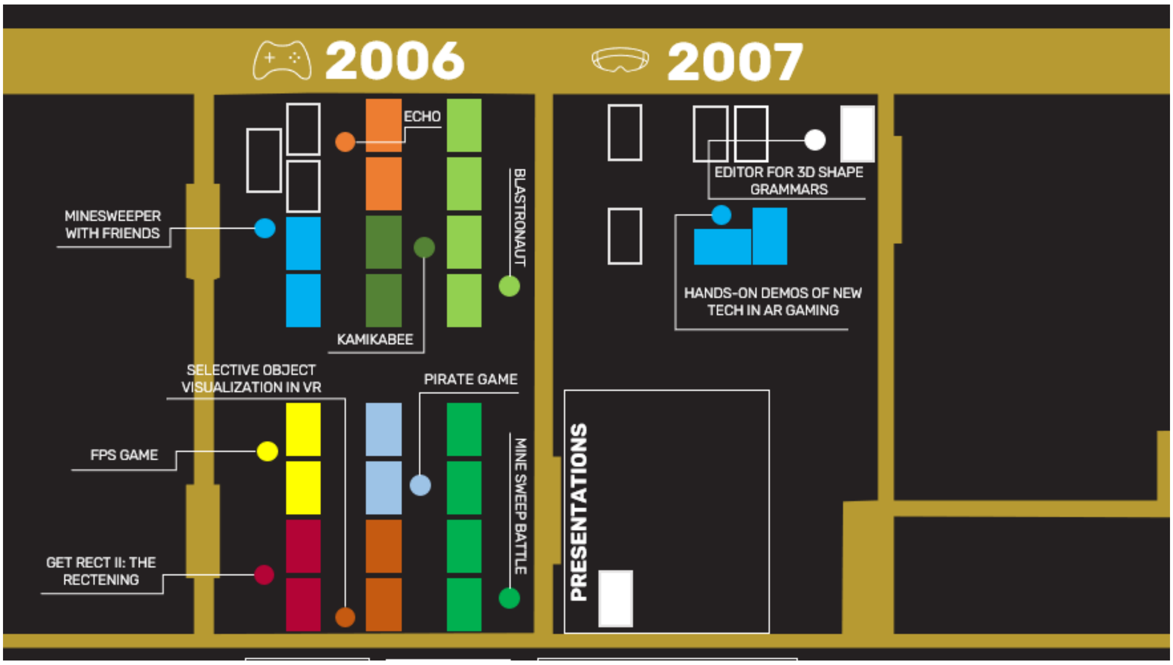

2006
2007
ECHO
BLASTRONAUT
EDITOR FOR 3D SHAPE GRAMMARS
MINESWEEPER WITH FRIENDS
HANDS-ON DEMOS OF NEW TECH IN AR GAMING
KAMIKABEE
SELECTIVE OBJECT VISUALIZATION IN VR
PIRATE GAME
FPS GAME
MINE SWEEP BATTLE
PRESENTATIONS
GET RECT II: THE RECTENING

Figure 27. Room setup for the second CGP Expo event.

Author: Daniel Nael.

V. License

Non-exclusive licence to reproduce the thesis and make the thesis public

I, **Kaarel Rüüsak**,

(*author's name*)

1. grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright, my thesis

Minesweeper With Friends – An Immersive Multiplayer Puzzle Game,

(title of thesis)

supervised by Mark Muhhin,

(supervisor's name)

2. I grant the University of Tartu a permit to make the thesis specified in point 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 4.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.
3. I am aware of the fact that the author retains the rights specified in points 1 and 2.
4. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Kaarel Rüüsak
15.05.2023