



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

*Distributed
Computing*



Mercenary Space Battle

Bachelor Thesis

Julian Andrea Croci

`crocij@student.ethz.ch`

Distributed Computing Group
Computer Engineering and Networks Laboratory
ETH Zürich

Supervisors:

Manuel Eichelberger

Prof. Dr. Roger Wattenhofer

June 11, 2018

Acknowledgements

I thank my supervisor Manuel Eichelberger for his assistance during the development of this game. During the weekly meetings we developed ideas and tested the game.

I thank all people who tested the game and did not get tired, even after they had to test out all possible strategies I considered to be viable playing the game.

Abstract

In this Bachelor's thesis, the goal was to create a game which implements the ideas of *asymmetric information* and *information hiding*. The result is a *2D twinstick shooter* named *Mercenary Space Battle* with two different game phases and some modifications which can be applied to the base game. The game can be played with two to four people.

In this written part the game play and special parts about the implementation are explained. Additionally, the motivation for this thesis and a discussion on the results is given.

Contents

Acknowledgements	i
Abstract	ii
1 Introduction	1
1.1 Motivation	1
1.2 Local Multiplayer	1
1.3 Information Hiding And Asymmetric Information	2
1.4 Related Work	3
2 Game Mechanics	4
2.1 General Mechanics And Goal	4
2.2 The Different Game Modes	7
2.2.1 Standard	7
2.2.2 Walls, Walls, Walls!	7
2.2.3 More Special Abilities	7
2.2.4 Cloudy	8
2.2.5 Battle Royale	8
2.3 Custom Settings	8
2.4 Menus	9
2.5 Tutorial	9
3 Implementation	11
3.1 Artificial Intelligence	11
3.2 Giving Players Feedback	12
3.3 The Art Of Mercenary Space Battle	12
3.4 Limitations Of The Unity Game Engine	13

CONTENTS	iv
4 Conclusion And Outlook	15
4.1 The Different Game Modes	15
4.1.1 Standard	15
4.1.2 Walls, Walls, Walls!	15
4.1.3 More Special Abilities	16
4.1.4 Cloudy	16
4.1.5 Battle Royale	16
4.2 Feedback From Play Testers	16
4.3 Outlook	16
Bibliography	18

Introduction

1.1 Motivation

Nowadays, video games are a widespread entertainment product being used by people across all ages. Some video games can be played together by several people, these games belong to the multiplayer category. Although most of the *multiplayer games* are played online, Games which can be played together at the same computer or console (called local multiplayer games) are still successful and in demand. The goal of this project is to explore new possibilities. Such a locally playable multiplayer game could incorporate the absence of information into the gameplay.

1.2 Local Multiplayer

There are two types of local multiplayer games: hotseat multiplayer Games and real-time multiplayer Games.

Hotseat Multiplayer Games

During a hotseat multiplayer game, only one person at a time will interact with the computer or the console. The principle is comparable to chess. One of the players will directly interact with the board at a time, the second one might think about possible moves but is not allowed to touch the board or the figurines. As in chess, the different players in a hotseat multiplayer game make their turns one after another.

Real-Time Multiplayer Games

During a real-time multiplayer Game, all participants control the computer or console simultaneously. Most of the time, this is realized by connecting multiple



Figure 1.1: Typical split screen multiplayer from *Halo*[1]

input devices such as gamepads or joysticks. It is comparable to a game of tag. All are active at the same time, running away from the catchers.

In both real-time multiplayer games and hotseat multiplayer games, for players it is quite hard to hide their current moves and strategies from the other players. This is, because other players might look at the part of the screen reserved for the interaction of one player with the game world as in games like *Halo*. Multiplayer games which are played on different machines connected via the internet, this problem does not exist as far as it is not intended by the creators of the game.

1.3 Information Hiding And Asymmetric Information

The terms *information hiding* and *asymmetric information* are very vague in the context of video game mechanics. Although their meaning seems to be intuitive, a definition for both terms which applies to this thesis is provided below.

Information Hiding

Information hiding means that information is not immediately visible to the player. The player can get the said information in some cases but he might get outdated information or has to pay a price for it. Information hiding only ever

occurs between the computer respectively the game and the player.

Asymmetric Information

Asymmetric information means that not all information a player has must be to the same extent be visible to the other players. This implies, that all players must have incomplete information.

1.4 Related Work

The game *Hidden in Plain Sight* [2] uses *information hiding* and *asymmetric information* in several game modes. From this game the idea that the player controls one of multiple characters which all look the same is used. Contrary to *Hidden in Plain Sight* a more complex game mechanic is used and the idea of trading its invisibility for having an advantage gets taken a step further. A classical board game whose gameplay is based on asymmetric information is *Cluedo* [3]. In *Cluedo*, the player solving a crime case the fastest wins. Every player has a tiny bit of information about the case, hence *asymmetric information* is involved. The players use questions to gather enough information to solve the case.

Game Mechanics

2.1 General Mechanics And Goal

The game is played with controllers and can be played with two to four persons. The players operate characters, which all look the same, on a two dimensional floor which is always displayed in whole on the screen. The “world” is limited by walls.

During the game, the player controls one character at a time. Characters are always part of a group of characters without logical or physical intersections as illustrated in Figure 2.1. The goal of the players is, that the character which they control does not die. Characters currently not controlled by a player are controlled by the computer and are called NPCs. The player can switch between different characters. The characters, controlled by the computer or the player, use weapons to eliminate other characters. The game is divided into two phases. During the *First Phase*, the players prepare for the *Second Phase* in which they will fight. The health of every character is shown on every character by filling one half of the character with 4 green *tiles*. As soon as the health of a character drops below 100 health points, the default amount of health, every 25 lost health points one tile disappears. An example is shown in Figure 2.2.

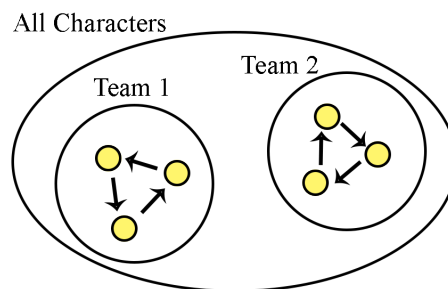


Figure 2.1: Illustration of the character divided into teams and the character switches a player can do indicated by arrows

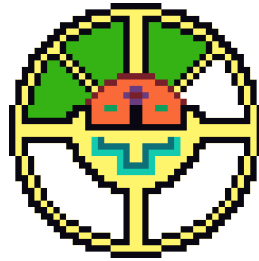


Figure 2.2: Character left with three health tiles

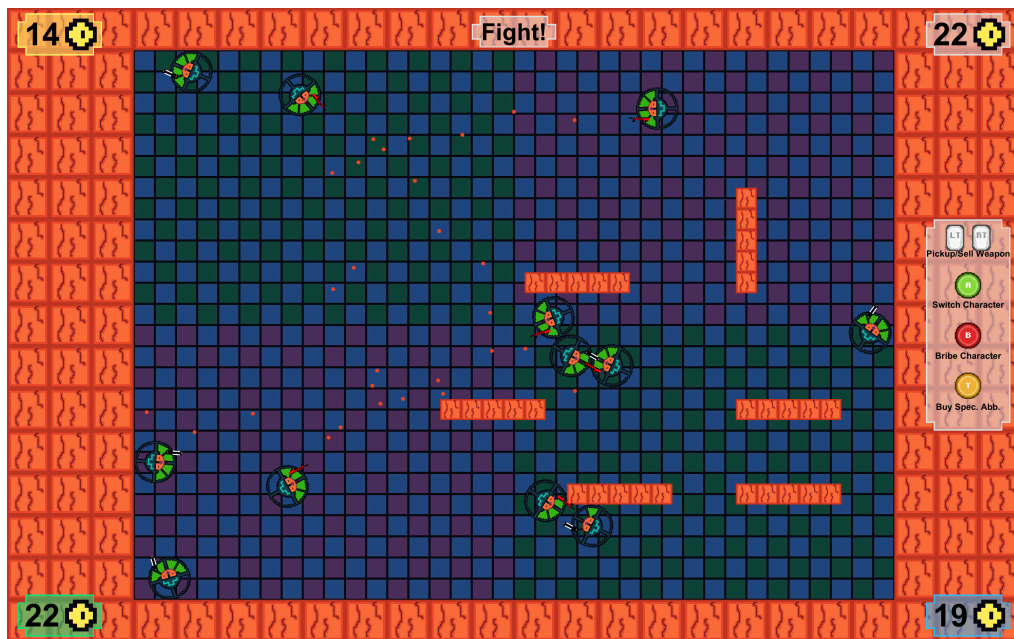


Figure 2.3: Screenshot of the *Second Phase* with four players

First Phase

The goal during the *First Phase* is to collect money, buy special abilities and bribe other characters.

At the beginning, the characters are divided into teams. Every player possesses a team, hence the number of teams corresponds with the number of players. Each player starts controlling a random character from their team.

During the *First Phase*, the players have the following options:

- Move/turn the character via the left/right joystick
- Switch to another character in their team by pressing A on their gamepad
- Pick up or sell one of their weapons
- Bribe a character not in their team to change teams in the *Second Phase* by pressing B
- Buy a special ability they can use during the *Second Phase* by pressing Y

Characters controlled by the computer only do the following:

- Move/turn in random time intervals
- Straight moving towards the nearest weapon on the ground
- Switch their left weapon to generate money for the team's player

When a character sells a weapon, buys a special ability or bribes another character, a short animation is shown. The money a player currently has is updated every five seconds in the corresponding corner of the screen.

The *First Phase* ends after a fixed amount of time, the default value is 60 seconds.

Second Phase

The goal of the *Second Phase* and at the same time the goal of the game is to eliminate all characters controlled by other players.

At the beginning of the *Second Phase*, all characters get assigned to their new teams and the special abilities of the players get activated. The remaining coins transform to a health bonus for the controlled character of the respective player. Depending on the selected game mode, additional things may happen.

In the *Second Phase* characters can only move around, turn and shoot.

The *Second Phase* ends when only one of the player controlled characters is still alive.

2.2 The Different Game Modes

Additionally to the standard mode, the game contains several modes altering or extending the above described core gameplay. The modes can be combined. If in the selection no mod is set, it is considered to be the standard mode.

2.2.1 Standard

The default gamemode comes with three special abilities *Spray And Pray*, *More Damage, Less Pew Pew* and *Bribing Bonus*. They can be bought during the *First Phase*. *Bribing Bonus* is activated when the player bribes a character from another team during the *First Phase*. The special abilities follow the idea of the rock paper scissors game. *Spray And Pray* is strong against *Bribing Bonus*, *Bribing Bonus* against *More Damage, Less Pew Pew* and *More Damage, Less Pew Pew* against *Spray And Pray*.

- *Spray And Pray*: This ability grants the player at the beginning of phase two two super weapons which can shoot multiple bullets at once. The downside of this ability is, that due to the special design of the super weapons the player loses its anonymity because he is easily identifiable by the other players.
- *More Damage, Less Pew Pew*: This ability transforms the weapons of the character at the beginning of the *Second Phase* to weapons which have the same look, instantly kill another character but can be shoot less often. Additionally, the player gets a big health bonus.
- *Bribing Bonus*: This special ability cannot be bought directly. It is activated when the player bribes another character during the *First Phase*. In the *Second Phase*, when the player dies, he switches automatically to a new character and so gets a second chance. He can do this as often as he bribed different characters in the *First Phase*.

2.2.2 Walls, Walls, Walls!

This mode works the same as the standard mode except that it starts with no walls on the battlefield. At the beginning of the *Second Phase*, a lot of walls spawn, possibly splitting the field into different areas. From there on every few seconds a wall disappears.

2.2.3 More Special Abilities

This mode works the same as the standard mode except that more special abilities exist. The following special abilities exist in addition to the special abilities

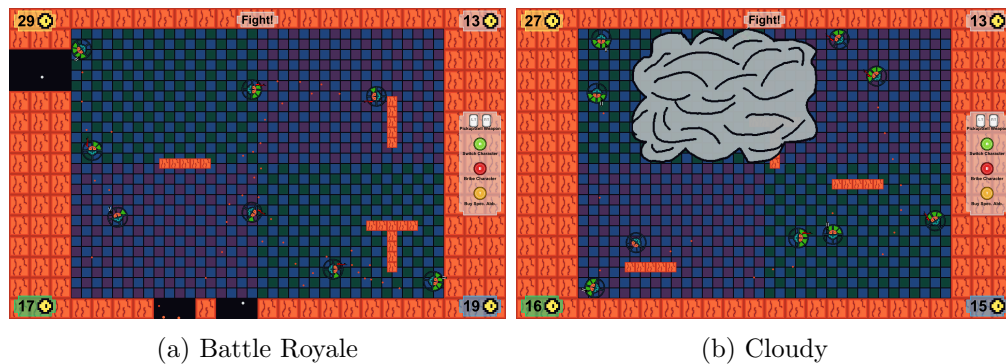


Figure 2.4: Game Modes

from the standard mode:

- *Increase Life*: Simply increase life at the beginning of the *Second Phase*.
- *Make All Others Big*: Double the size of all other characters.
- *Shrink When Hit*: Reduce the size of your character linearly as soon as your health drops below 100.

2.2.4 Cloudy

This mode works the same as the standard mode except that in the *Second Phase* a cloud spawns. The cloud moves over the battleground covering around a quarter of the area. A screenshot is shown in Figure 2.4.

2.2.5 Battle Royale

This mode is inspired by the popular action game genre *Battle Royale*. In a *Battle Royale* game, the space where players can be without suffering negative effects gets smaller with time. In this game mode, walls disappear and create a suction sucking characters out of the spaceship as shown in Figure 2.4. Characters outside the spaceship die. The effect area of these suctions grows larger with time.

2.3 Custom Settings

Players can load custom settings from a file they have to place at the same place in their file system as the executable. The typical content of such a file could be the following:

```
CharactersPerTeam=5
StatusRefreshTime=5.0
CharacterSwitchBackoff=10.0
DurationPhase1=60
```

`CharactersPerTeam` controls the number of characters per team and can be a value between two and six. `StatusRefreshTime` gives the interval in which the money display for every player is updated, this can be any positive float. `CharacterSwitchBackoff` is how long a player has to wait before he can switch the character again during *First Phase*. Per default this is a very small value but it can be any positive float. `DurationPhase1` is the duration of phase one, this can be any positive integer.

2.4 Menus

All the menus in the game can be controlled with a controller. Before the actual game can start, all players have to register their controller by pressing the Y button on their controller. When they pressed the button, a new character will be shown in this player registration screen with their color. If they press the button again, their character becomes black so that they can find out which their character is when multiple persons register their controller simultaneously.

2.5 Tutorial

For learning the basic controls of the game for moving, shooting, buying special abilities, switching and selling weapons, a tutorial is accessible from the main menu. The tutorial can only be played by one person at a time and is just a short sequence of different tasks like moving the character or shooting an enemy. A screen shot of the tutorial is shown in Figure 2.5.

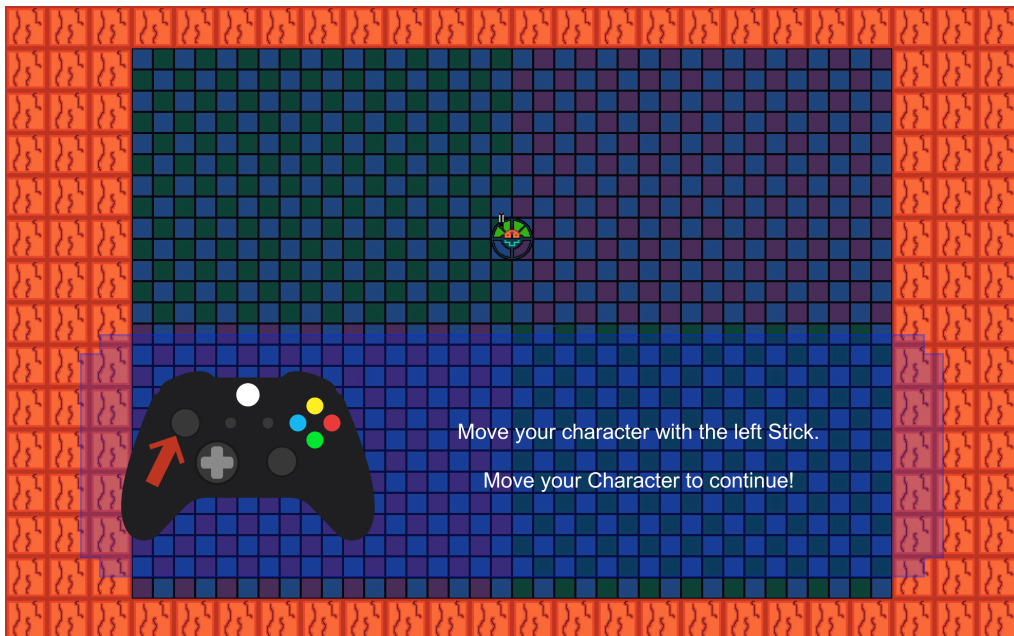


Figure 2.5: Tutorial Screen

Implementation

Mercenary Space Battle is a purely 2D game. The game was developed with the Unity3d game engine Version 2017.2 [4]. The engine handles the physics calculations and the drawing of the sprites.

3.1 Artificial Intelligence

The AI is not very advanced mostly because it should be possible to distinguish the characters controlled by a player and the characters controlled by the AI. For example, the AI ignores obstacles on the battlefield in its pathfinding or does not dodge bullets flying towards the character.

In the *First Phase*, the AI controlled characters move with a given chance to the next weapon. Otherwise, they will just move into a random direction for some amount of time. The AI only switches its left weapon.

In the *Second Phase*, the AI shoots and aims at targets. After some time the AI might change its target. The AI considers the closest two characters as target. There is a small chance that the character will target a character in the same team as well that it will not target a character from the enemy team. If the NPC does not aim at a other character, it just moves around randomly as in the *First Phase*. Aiming works by changing the orientation according to the targets location (with some random error) every random amount of milliseconds.

To account for the holes in the walls in the *Battle Royale* mode (2.2.5), the AI has two basic states influencing its movement in the *Second Phase*. State one: Not near a hole in the wall. State two: Near to one or multiple holes in the wall. If the character is in state one, it will just move into a random direction. If the character is in state two, it will move in the opposite direction of the holes. The direction of the movement of the character is in this case defined as $\sum_{\forall \vec{X}_i} \vec{Pos} - \vec{X}_i$

where \vec{X}_i is the position of a hole nearby and \vec{Pos} is the current position of the character. A state is changed when the character moves into the collider of a

hole or if it leaves the collider of a hole. A counter is used to keep track of the number of holes the character is near to. The references to the hole-objects are saved in a linked list.

3.2 Giving Players Feedback

Although the game is about *information hiding*, it is important that the player has some sort of feedback if he buys a special ability or sells a weapon. The favourite solution would have been to signal a successful transaction by letting the controller of the player vibrate. Sadly, this is not possible with the current *Unity* [4] version for all operating systems. As an alternative, the players coin display wiggles and directly on the character is a short animation played.

3.3 The Art Of Mercenary Space Battle

For this game, I created most of the sprites myself, mostly with *Adobe Photoshop* [5]. The sprites were created with retro pixel graphics as in *Super Mario Bros.* [6] or more recent *Stardew Valley* [7]. The only sprites I did not create are the gamepad button [8] and the gamepad [9] in the tutorial. The color palette comes from the upcoming videogame *Starmancer* [10]. All Graphics in the game consist of colors from this palette except the gamepad buttons and the gamepad.

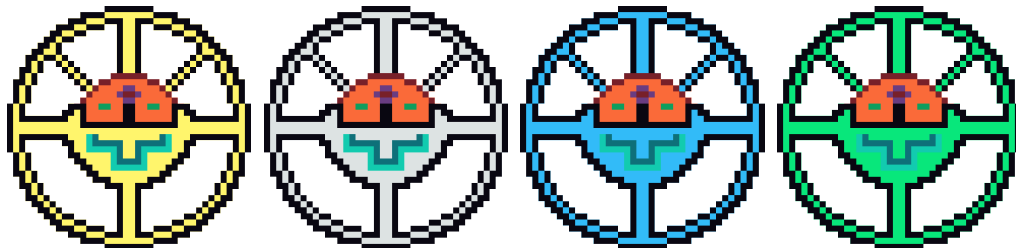


Figure 3.1: The character sprites in all four colors

The game is built with a grid spacing of around 40 pixel per unit to be able to scale it evenly for multiple screen resolutions. The battleground itself always has the same dimension, namely 18 times 13 units. Due to the goal to have pixel perfect sprites on the screen and constant size of the battleground across multiple resolutions, the size of the border walls is increased or decreased depending on the screen resolution as shown in Figure 3.2. The walls are created by multiplication of a 40 times 40 pixels tile. The floor consists of a single graphic because its size is constant but it follows the idea of tiles. The character graphics are based on a circle with a diameter of 40 pixels.

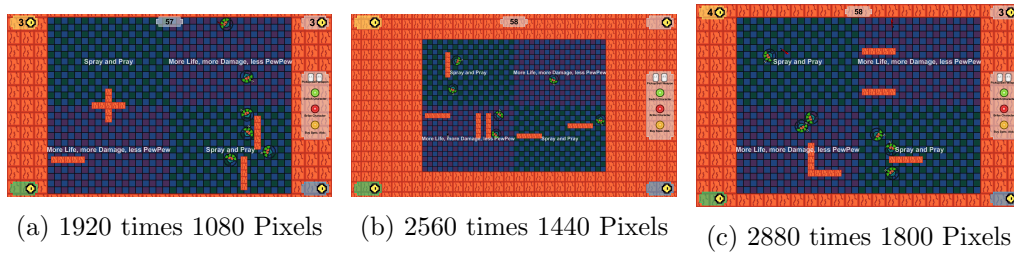


Figure 3.2: Comparison of different screen resolution and the different thicknesses of the walls



Figure 3.3: Text in Menu with small pictures within

3.4 Limitations Of The Unity Game Engine

Although *Unity3d* [4] is a beginner friendly engine and getting into it is quite easy, some things are quite unintuitive, some features absent. This was especially noticed during the design and implementation of the UI.

Although Unity can be used to develop for almost every current console, creating menus accessible for controllers is quite hard. Without the shared knowledge of the Unity community and some hacks, it would not have been possible to create the menus in the game to be fully navigable with the controller. Especially the dropdown menu provided by Unity is not usable with a controller. A custom script is needed to manually scroll the list in the dropdown menu to the position of the currently selected option.

Another missing feature is the option to include small pictures into some UI-Text. Sure, the possibility to create one's own typeset exists, but it seems to be quite a hardship. For this game, the problem is solved by putting an appropriate amount of spaces into the text and placing the small symbols at the appropriate position over the text container. The disadvantage of this approach is, that if the text has to be changed after positioning the symbols, the symbols need to be repositioned. Additionally the positioning of the symbols might be a little bit wrong because blanks around the symbols are quite hard to get correct manually. An example is shown in Figure 3.3.

Conclusion And Outlook

4.1 The Different Game Modes

Although the game modes are freely combinable in the following section the game modes will be evaluated separately.

4.1.1 Standard

The main idea of the *Standard* mode is to deliver a game having three different strategies, namely the three special abilities being balanced. All in all the information hiding works quite well. For two of the three special abilities it makes really sense to hide which character one plays, at least at the beginning of the *Second Phase*. This can be achieved by switching characters often during the *First Phase*. None the less the game mode is by far not the most fun to play of all modes. As well, the balancing is not quite on point currently. The special ability *Super Weapon* is too strong and the best way to win. At the end of phase two, it is possible, that two player have a really long battle because they can hide both behind walls without pressure to take initiative.

4.1.2 Walls, Walls, Walls!

Although the same special abilities are used as in the *Standard* mode, the strength of the different special abilities changed. Especially *Super Weapon* is less strong while *More Damage*, *Less Pew Pew* is more effective. This is because the segmentation of the field prohibits that the players with *Super Weapon* kill a lot of characters quickly. All in all the *Second Phase* is way slower because most players have only few characters they can even reach at the beginning. This gives the players way more time to plan their actions.

4.1.3 More Special Abilities

This mode is mostly a showcase for all special abilities created during the development of the game. It is not balanced by any means.

4.1.4 Cloudy

Due to the impaired vision on the field, this game mode takes the idea of information hiding a little further. Other than the *Walls, Walls, Walls!* mode, the performance of the different strategies does not change much. *Super Weapon* is still the best due to the possibility to just shoot in to the cloud or shoot out of it without being seen.

4.1.5 Battle Royale

As well as the *Cloudy* mode, another layer of information hiding is given due to the fact, that the scope of the holes is not shown. Additionally, possible long lasting duels at the end of phase two are avoided. This is, because the area the holes affect grows. The space the players can be in safely gets smaller. After some time, the players have to act. If the players get into the area of effect of a hole, they have to focus on not getting sucked out of the ship. This gives the competitor a chance to start an attack.

4.2 Feedback From Play Testers

During and near the end of the development, the game was tested by some friends. Their favourite game mode was by far *More Special Abilities* due to the additional and in their eyes funnier special abilities as for example *Shrink When Hit*. Another interesting observation was, that they most of the time just spun the character they controlled around so that they could find it although this would reveal them to the other players. The reason for this might be, that different to *Hidden in Plainsight* [2], the main focus of this game might be more on the fighting and less on the basic idea of staying hidden. In this game, revealing themselves is way more an option if not a necessary evil with some special abilities.

4.3 Outlook

All in all, the game works and offers enough variety to be interesting enough for one or two hours even for persons which do not play video games occasionally. At the moment, the look of the game is quite simplistic. More varied art and

sprites for the backgrounds could improve the game as well as more and better animations. Although some simple animation infading and outfading walls exist, which are only around for aesthetic reasons, the more complex animations, which are for example played when a character sells its weapon, exist mostly to give the player feedback. Gameplay wise, the different special abilities need a better balancing.

Bibliography

- [1] Bungie: Halo: Combat Evolved. Videogame (2001) Screenshot captured on the Masterchief Collection for xBox One [Online; <https://www.halowaypoint.com/>].
- [2] Spragg, A.: Hidden in Plain Sight (2014) [Online; https://store.steampowered.com/app/303590/Hidden_in_Plain_Sight/].
- [3] Pratt, A.E., Pratt, E.: Cluedo (1948)
- [4] Unity Technologies: Unity3d game engine [Online; <https://unity3d.com>].
- [5] Adobe: Adobe Photoshop [Online; <https://www.adobe.com/products/photoshop.html>].
- [6] Nintendo: Super Mario Bros. (1985)
- [7] Barone, E.: Stardew Valley. Published by Chucklefish for PC, PS4, xBox One, Nintendo Switch (2016) [Online; <https://stardewvalley.net>].
- [8] HeyHumans: xbox 360 pixel art controller icons License CC-BY-SA 3.0 [Online; <https://opengameart.org/content/xbox-360-pixel-art-controller-icons>].
- [9] Max Pixel: Xbox computer game video games controller gamepad License CC0 1.0 [Online; <https://www.maxpixel.net/Xbox-Computer-Game-Video-Games-Controller-Gamepad-1827840>].
- [10] Ominux Games: Starmancer [Online; <https://www.playstarmancer.com>].