



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

*Distributed
Computing*



Hidden in Plain Sight

Semester Thesis

Maxim Aebischer

`maxima@ethz.ch`

Distributed Computing Group
Computer Engineering and Networks Laboratory
ETH Zürich

Supervisors:

Manuel Eichelberger, Michael König
Prof. Dr. Roger Wattenhofer

March 31, 2017

Acknowledgements

I thank my supervisors Manuel Eichelberger and Michael König for their assistance during this project. They took the time for our weekly meetings and gave me very helpful feedback when we tested the game together. I would like to thank them for the many ideas they had considering the game.

I thank all my friends and family who have tested the game. I would like to thank them for their feedback that helped a lot improving the game.

Abstract

In this project, I created a local multiplayer video game which implements the principle of asymmetric information. The game is a 2D-platformer for 2-4 players. At the beginning of the game the player starts with no information at all. He has to gain information about the other players while hiding himself.

In this paper, I will present how asymmetric information is used in the video game. Furthermore the paper shows what the idea behind the game is and how the most important functions of the game were implemented. The game has many AI controlled characters and I will present how these characters are controlled.

Contents

Acknowledgements	i
Abstract	ii
1 Introduction	1
1.1 Motivation	1
1.1.1 Video Games	1
1.1.2 Asymmetric Information	2
1.2 Related Work	2
2 Game Concept	3
2.1 Goal	4
2.2 Game Engine and Settings	4
3 Implementation	5
3.1 Controllable Character	5
3.1.1 Human Controller	6
3.1.2 AI Controller	7
3.2 Game Manager	8
3.3 Upgrade System	9
3.4 User Interface	9
3.4.1 Heads-Up Display	10
4 Gaming Experiences	12
4.1 Strategies	12
4.1.1 2 Player Strategy	12
4.1.2 3 or 4 Player Strategy	12

CONTENTS	iv
5 Conclusion and Outlook	14
5.1 Outlook	14
Bibliography	15

Introduction

1.1 Motivation

1.1.1 Video Games

Video games are nowadays as popular as they have ever been.[1] Video game sales are higher than ever before. And specially multiplayer games have become more popular over time.

The first multiplayer games were created in the 1970s as arcade games. One of the first popular multiplayer games was Pong by Atari. Later the multiplayer gaming industry really took off with the introduction of gaming consoles such as the Nintendo 64 or the Playstation. They introduced the concept of multiple controller ports which offered the possibility that multiple people could play on the same console. This started the trend of local multiplayer games, also called couch games. Games such as GoldenEye 007 or Mario Kart were sold millions of times[2][3].

In the 2000s online gaming began getting popular. With increased internet accessibility and speed, many started playing video games over the internet. Here players use the internet to connect to each other instead of being in the same room. A big advantage of online games is that it is easy to find people to play with. You can basically play with everyone around the world.

Lately there has been a trend of going back to local multiplayer gaming. Games such as Wii Sports (2006), the LEGO series (2007-), TowerFall(2015) or GangBeasts(2014) were financially successful local multiplayer games in the last few years. Even though most multiplayer games and gaming consoles focus on online gaming, there are some successful games that chose to incorporate offline multiplayer gaming.

1.1.2 Asymmetric Information

Asymmetric information is defined as a type of information distribution where some players have different information available to them. Basically some players may know more than other players.[4] Asymmetric information requires that at least one of the players has imperfect information. This means that one aspect of information about the game is not known to all the players. The information can be faulty or the players may simply lack of that information. Often players also gain information that they are not certain about. They may act differently even though they have the same information. This also leads to asymmetric information.

Asymmetric information is often balanced between all players. In this case, players have the same abilities to gather information. But because the information is imperfect or they are uncertain about it, there still is asymmetric information. This is used in many card games, such as in poker, where each player does not know what all the other players have in their hands. This is also referred to as potential symmetric information. All players could potentially figure out the distribution of cards by analysing actions and events throughout the game. So they could potentially achieve perfect, symmetric information.

However asymmetric information can also be unbalanced. In many guessing games one player knows the answer and has perfect information, while all other players need to guess this answer and therefore have imperfect information. Such an unbalanced game is the word-guessing game charades. The player that knows the answer will act out a word and give some information to the other players, who then have to guess based on their imperfect information.

1.2 Related Work

A video game that tries to incorporate the principle of asymmetric information is Hidden in Plain Sight by Adam Spragg[5]. This game has many different modi. In some the players have entirely different goals to achieve and different abilities. These modi use the principle of unbalanced asymmetric information. In other modi the players start with no information at all and have to gain information to identify the other players among artificially controlled characters. These use balanced asymmetric information.

Game Concept

The main goal of this project was to incorporate the principle of asymmetric information in to the video game. In this game we use balanced asymmetric information. The players start with no information at all and have to gain the information themselves. Each player controls one of many characters and the rest is controlled automatically. The players first have to identify their own character and then try to identify those of the other players. Because we have so many characters on the play field, the player can never be certain that the information they gain is correct and will rarely achieve perfect information.

Furthermore the game is a local multiplayer game. It is played on a single screen that displays the whole playing field. Moreover the game is a so called platformer. The player can walk horizontally and jump around to reach the different height levels. The background consists of a ground floor and many houses of different size. The houses are the platforms of the game. The platforms have a one directional collision detection. This means the player can jump through the platforms but can not fall down.

The game can be played with 2 to 4 players. This can be chosen at the start of the game. This option was implemented because you may not always have 3 friends to play the game.

The game starts with many characters on the ground level[fig. 2.1]. We have 10 characters that are controlled by an artificial intelligence and up to 4 characters controlled by the players. These characters have all the same look and are not distinguishable. So the player does not know which one of the characters he is himself and which ones are controlled by the other players.

Moreover there are different treasures distributed on the playing field. These treasures contain one of three resource that the characters can collect. If a character walks through a treasure, he will get one resource allocated to that treasure. Every one of the three resource can be collected at three different treasures. So we have 9 treasures in total. These treasures are distributed on the ground floor and the platforms[fig. 2.1].

Once the players have collected enough resources, they can use them to up-



Figure 2.1: The start of a round. All characters are on the ground and not distinguishable. The countdown is running.

grade their character. The character can at most have 5 pieces of each resource.

2.1 Goal

The main goal of the game is to survive the longest out of all human players. To achieve that the player has to collect the different resources to buy and upgrade his gun. He can then fire bullets that inflict damage on the players they hit.

The player that survives the longest wins the round and will get one point. The players do not get points for destroying other characters. They only get a point if they are the last living player of that round. The players can play as many rounds as they want. The points will simply added up every round.

2.2 Game Engine and Settings

The game was implemented using the Unity3D game engine[6]. This engine handles the graphics and physics of the game. Scripts for the different game objects were written with CSharp.

At the start of the game you have the option to choose the resolution of the game and to set the input bindings. The The input controls are pre-set for four gamepads. Additionally you can control the first two players with your keyboard. You could also set the inputs of the other players on the keyboard, however every player needs at least 7 buttons to play the game.

Implementation

3.1 Controllable Character

Non-playable characters (NPC) and those controlled by the players are based on the same controller. Up to four playable characters are controlled with a gamepad connected to the computer on which the game is running. All non-playable characters are automatically controlled by an artificial intelligence (AI). All characters have the same look[fig. 3.1]. They are based on the Unity3D demo project "2D Platformer".[7]



Figure 3.1: The controllable character based on [7]

The characters have 100 health points. They lose health points every time they are hit by a bullet. Once the character loses all his health points, he will be destroyed. Furthermore the controllable characters have many different functions.

To move the player horizontally the controller adds a force to the character in the chosen direction. The Unity game engine will then compute the velocity

of the player accordingly. The controller has a maximum velocity which can not be exceeded. Additionally if the player walks out of the screen, he will be placed on the opposite side. That means if a character tries to exit the screen on the left side, he will enter again on the right side.

To move vertically the player is able to jump. If the player is grounded, either on the white floor or on one of the platforms, he can jump with a given velocity. He will then reach a certain height based on his jump velocity and the gravity.

If the character hits a treasure he will get a resource based on the treasure type. There is no actual collision. That means the player can simply walk through the treasure to gain a resource. The player can then use these resources to buy and upgrade his gun, armour or health. The player can also heal himself. If he has already lost health points, he can add some more for the price of one resource each.

As mentioned before the player can buy a gun. He can then shoot this gun. If the player chooses to shoot, a bullet will be instantiated at his position. This bullet will then move horizontally in the direction the player is facing. If the bullet collides with another player, the player hit by the bullet will lose health points according to the damage of the shooters gun.

An important feature of the controller is the split. For the price of two resources each the player can instantiate a copy of himself. This new character is controlled by a new instance of the AI controller. The clone has the same upgrade values, health points and resources as his originator.

The controllers are reset before every round of the game. During the reset all upgrades are deleted, the health points and resources set on their initial value and dead players are revived again.

3.1.1 Human Controller

As mentioned before the playable characters are controlled with a gamepad. The inputs of the gamepad are handled by the human controller. The following input bindings are described for a Xbox 360 gamepad[fig. 3.2]. These bindings may differ a bit for different gamepads.

The player can move horizontally with the left joystick of his gamepad. The horizontal value of this joystick is then used by the controller to add the move force to the character. If the player is grounded and he presses the A face button on his gamepad, his character will jump. With the B button of the Xbox 360 controller the player can shoot his gun. This only works if the player has bought the gun already. To heal himself the player can press the X button. This only works if the player has enough resources and his health points are not full. The Y button is bound to instantiate a copy of the player. This also needs enough resources to work.

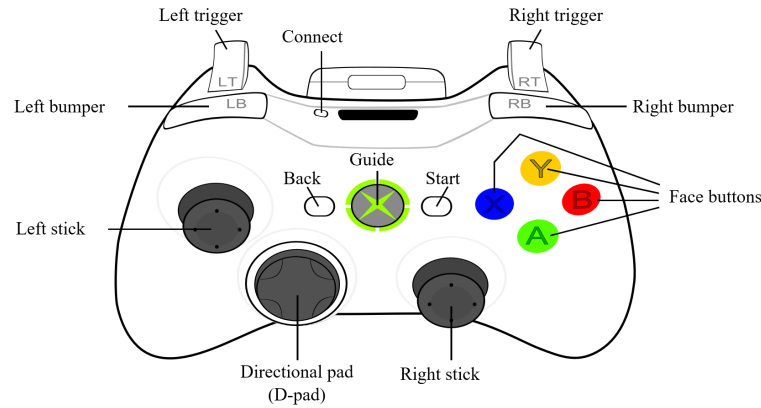


Figure 3.2: The Xbox 360 controller

The human controller can use the upgrade functions with the help of the left bumper of the gamepad. The combination of the left bumper and the jump button will try to upgrade the armour of the player. In combination with the shoot button the controller buys or upgrades the gun of the player. And lastly the player can upgrade his health with the combination of the heal button and the upgrade button.

The player has also the option to pause the game. If he presses the start button of his gamepad the game will be paused and the menu screen activated.

3.1.2 AI Controller

The non-playable characters are controlled by the artificial intelligence controller. The AI controller is based on the normal controller but it has some additional functions. Most importantly the AI controller has different AI states which control what the AI controller does next. These states are described subsequently.

The first and most simple AI state is the walk state. If the AI controller is in this state he chooses randomly a position on the screen to which he controls the player to go to. Once the player reaches this position he chooses the next position or changes the AI state with a given change probability. If the horizontal distance between the actual position and the next chosen position is greater than half the screen width the controller will exit the screen in the wrong direction and enter again on the opposite side.

The next state is the collect state. This state is similar to the walk state with the difference that the position to choose from are the positions of all existing treasures. This means that the player will walk from one treasure to the next. These two states will only move around and jump. They will never upgrade or shoot the gun.

The upgrade state chooses one of three upgrade strategies. These strategies are "gun", "armour" or "health". The controller will then collect the resources he needs to upgrade according to his strategy. When the controller upgraded he will change his state or choose a new strategy. Again this decision is made with a pre given probability.

There is also a mimic state. When the controller is in this state he chooses another controller to mimic. He then tries to upgrade his gun, armour and health to reach the same level as the controller he mimics. In this state the controller changes his state only when he has the same upgrade levels as the controller he mimics.

Lastly there is the shoot state. In this state the controller chooses a target and walks around. Once the target is on the same height as he is himself, the AI controller fires his gun. The controller shoots a bullet roughly all 0.2 seconds. He will change his state or target with a given probability after every shot.

Every AI controller is always in one of those states. The probabilities to choose the AI state are set identically for every AI controller. However they can be set during the game. This is only done when the player splits himself. The cloned character will stay in the mimic state and copy the player as long as he is alive.

3.2 Game Manager

The game manager handles everything around the actual game. If you start the game he will deactivate the human controllers that are not needed. The manager will inform the user interface of the number of human players so that the interface only displays actual players. The manager also sets all characters to a start position. These are randomly chosen positions on the ground. Before the round can start, the manager starts the start countdown during which the controllers are disabled and the players can't move.

During the game the manager pauses the game if one the players hits the start button. He disables all controllers and displays the pause menu. This means no character can move during the pause. When the player chooses to resume the game the manager will start a countdown before all controllers are enabled again. The manager also counts the number of living players. If there is only one human controller left, the manager will end the round, add a point to the surviving player and show the current ranking.

When the players choose to start a new round the manager resets all controllers and destroys all bullets and clones of the last round. The manager then starts a new game as before.

3.3 Upgrade System

The player can choose to upgrade three different things during the game. All upgrades cost a certain amount of resources which is shown on the heads-up display during the game. Each upgrade direction has four different upgrade levels and the costs are increased with every level.

The first thing to upgrade is the gun of the player. With the first upgrade level the player buys the gun and gains the ability to shoot. The gun has a basic damage value of 5 health points. This means if a bullet of a gun with level 1 hits a player he will lose 5 health points. The gun can be upgraded up to level 4. The damage of the bullets increases with every level. The maximum damage is 300 health points.

It is also possible to upgrade the armour of the player. With every upgrade level the damage taken by a bullet will be decreased with the damage multiplier. A player with no armour has a damage multiplier of 1. The multiplier with armour level 4 is only 0.05 which means that a player with fully upgraded armour can easily survive being hit by a bullet of a fully upgraded gun.

The color of the gun and the armour changes with every upgrade. This means that every player knows the upgrade levels of every character and the players have to decide carefully, when they want to upgrade their character. A player with a high upgrade level may be identified and attacked more often.

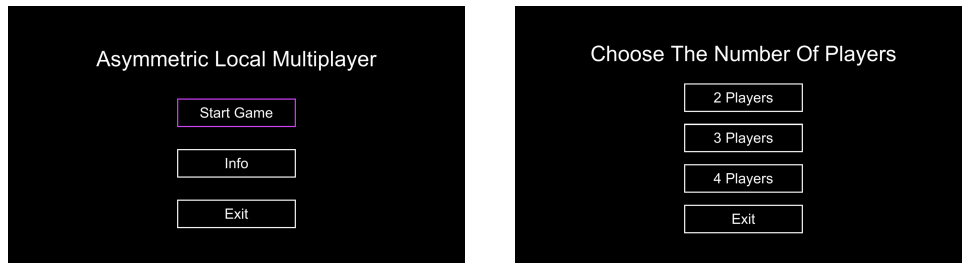
The last thing to upgrade is the health of the player. With the first upgrade the player gains the ability to heal himself. He will gain 10 health points when he tries to heal himself. With every further upgrade the health points that the player can gain are increased. At health level 4 the player will gain 100 health points every time he heals himself.

3.4 User Interface

The user interface is kept simple but has all necessary functions. The game starts with the main menu.[fig. 3.3a] The players have the possibility to start a game, look at the controls or exit the game. If the player chooses to start a game he will be asked to select the number of players before the game actually starts.[fig. 3.3b]

During the game the players have the choice to pause the game. This will display a pause menu.[fig. 3.4a] In this menu the players have the possibility to show the current statistics. This stats menu shows the current resources and points of each player.[fig. 3.4b] Furthermore they can check the controls, resume the game or exit the game and return to the main menu.

At the end of each round the current ranking is displayed and the players



(a) The main menu

(b) The start game menu

Figure 3.3: The interface at the start of the game



(a) The Pause Menu

(b) The Stats Menu

Figure 3.4: The user interface if you pause the game and the current statistics

can choose to start a new round or return to the main menu.[fig. 3.5]

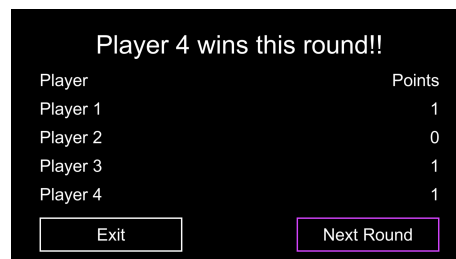


Figure 3.5: The interface after a completed round with the points of every player

The players can move through the menu buttons with the left joystick and select the buttons with the jump button of their gamepad. Alternatively they can also select the menu buttons with the mouse.

3.4.1 Heads-Up Display

During the game the heads-up display (HUD) is shown.[fig. 3.6] Because it should be difficult to detect other human players, only few things are shown on the HUD. On the bottom of the screen the upgrade costs and values are shown. The colors



Figure 3.6: The heads-up display

of each upgrade level corresponds to the color of the gun and armour at that level. Furthermore on the top of the screen it is displayed if the players still live and which health level they have. If the player is dead, his player icon will get transparent.

The current resources and the two other upgrade levels of the player are intentionally not shown. It would give the players too much information about the others if they were shown.

Gaming Experiences

The feedback to the game was mostly positive. Most of the people who played the game said that the concept of not knowing anything at the beginning of the game was new to them. That they had to gather the information about the other players themselves was a feature they enjoyed a lot. The games could get very emotional as well. It happened a lot that players thought that they have destroyed another player, only to find out that it was an AI controlled character.

4.1 Strategies

The game strategies are very dependent of the number of players in the game. It is a huge difference if you play the game with only one adversary or if you play it against multiple people. Because of that difference we will discuss these two cases separately.

4.1.1 2 Player Strategy

If you play the game with only two human players the best strategy to win the game is to upgrade your gun as fast as possible. It does not matter if the other player can detect you or not. As long as you have the better upgrades than your adversary, you will most likely win the game. This means the game becomes a simple shooter game. You just have to get a good gun and shoot the other player before he will shoot you. A defensive strategy, meaning that you try to act like an AI controlled player and not showing yourself, is not as effective in a 2 player scenario because you still have to shoot your adversary yourself.

4.1.2 3 or 4 Player Strategy

The game is a lot more interesting if you play it with 3 or 4 players. In this scenario the players who are not identifiable by the others have a much higher chance to win the game. This means that you can also win the game with a

defensive strategy. A player that has good armour and health upgrades has a good chance to survive for a long time, even if his gun is not as good as the others.

Often the players who were aggressive in a 3 or 4 player match and tried to shoot the others first, were the ones to die first themselves. As soon as the first player starts shooting others, all other players can pretty easily identify him and often tried to destroy him together.

An important thing in a 3 or 4 player match is also the splitting. Once you have used your gun to destroy one player you will be easily identifiable by all the other players. When you are able to split yourself you make it difficult for the others to still find you because there is for sure an exact copy of yourself somewhere on the screen. So there is a chance to become anonymous again.

Conclusion and Outlook

The goal of this project was to create a local multiplayer video game which implements the concept of asymmetric information. We created a multiplayer video game in which the players start with no information at all and then try to gain the needed information about the other players. However if the players use their possibilities well it is very difficult to be sure which players are human controlled and which ones are artificially controlled. The game works fine on a Windows PC and the feedback to the game was mostly positive.

For 3 or 4 players the strategy of acting like an AI controlled character is pretty effective. The game is then pretty well balanced between the different strategies. However in the 2 player case the game often becomes a shooter game. It is too easy to win the game without trying to hide yourself.

5.1 Outlook

The game is still far from perfect. The AI controllers have only few basic functions and the characters controlled by them are pretty easy to detect. It would be very interesting to have AI controllers who are more "human". They do not have enough flaws yet. For example, as soon as a character makes a pause, everyone can be certain that it is a human player.

The balancing of the game could be improved as well. In a 2 player game, it is too easy to win without hiding yourself. It also does not matter if you hit AI controlled characters. This problem could be avoided by implementing some kind of punishment if a player shoots too much. Maybe a limited number of bullets or negative points when you kill a NPC helps with that.

Most importantly the game does not look very interesting. With better graphics and animation it would be a lot more fun to play the game. The background does not have any texture at all and the character has no animation for most of his actions.

Bibliography

- [1] Wikia, V.G.S.W.: Video game industry (2017) http://vgsales.wikia.com/wiki/Video_game_industry [Online; accessed 31 March 2017].
- [2] Gameinformer: Mario kart sales (2014) <http://www.gameinformer.com/b/news/archive/2014/06/02/mario-kart-8-speeds-to-over-1-2-million-sales-in-opening-weekend.aspx> [Online; accessed 31 March 2017].
- [3] VGChartz: Goldeneye 007 (2017) <http://www.vgchartz.com/game/877/goldeneye-007/> [Online; accessed 31 March 2017].
- [4] Bjork, S., Holopainen, J.: Patterns in Game Design. Charles River Media game development series. Charles River Media (2005)
- [5] Steam: Hidden in plain sight (2014) <http://store.steampowered.com/app/303590/> [Online; accessed 31 March 2017].
- [6] Unity3D: Unity 3d game engine (2017) <https://unity3d.com/unity> [Online; accessed 31 March 2017].
- [7] Unity3D: Unity3d 2d platformer sample project (2017) <https://www.assetstore.unity3d.com/en/#!/content/11228> [Online; accessed 31 March 2017].