Artificial Intelligence in Games
Faking Human Behavior

Mattias Edlund

Faculty of Arts
Department of Game Design
Bachelor’s Thesis in Game Design, 15 hp
Program: Game Design and Programming
Supervisors: Mikael Fridenfalk, Hans Svensson
Examiner: Masaki Hayashi
June, 2015
Abstract

This paper examines the possibilities of faking human behavior with artificial intelligence in computer games, by using efficient methods that save valuable development time and also creates a more rich experience for the players of a game. The specific implementation of artificial intelligence created and discussed is a neural network controlling a finite-state machine. The objective was to mimic human behavior rather than simulating true intelligence. A 2D shooter game is developed and used for experiments performed with human and artificial intelligence controlled players. The game sessions played were recorded in order for other humans to replay. Both players and spectators of the game sessions left feedbacks and reports that could later be analyzed. The data collected from these experiments was then analyzed, and reflections were made on the entire project. Tips and ideas are proposed to developers of shooter games who are interested in making human-like artificial intelligence. Conclusions are made and extra information is provided in order to further iterate on this research.

Keywords: Artificial Intelligence, AI, games, human behavior, neural network, AI in games
**Sammanfattning**


**Nyckelord:** Artificiell Intelligens, AI, spel, mänskligt beteende, neurala nätverk, AI i spel
Table of Contents

1 Introduction ................................................................................................................................. 1
  1.1 Background .............................................................................................................................. 1
    1.1.1 Artificial Intelligence ......................................................................................................... 1
    1.1.2 Human Brain Project ......................................................................................................... 1
    1.1.3 Turing Test ....................................................................................................................... 2
    1.1.4 Previous Research ............................................................................................................ 2
  1.2 Definitions .................................................................................................................................. 3
    1.2.1 Game .................................................................................................................................... 3
    1.2.2 Game Engine ..................................................................................................................... 3
    1.2.3 Artificial Intelligence ......................................................................................................... 4
    1.2.4 Bots ..................................................................................................................................... 4
    1.2.5 Neural Network .................................................................................................................. 4
    1.2.6 Finite-State Machine ......................................................................................................... 5
    1.2.7 Remote Procedure Call ..................................................................................................... 5
  1.3 Problem Statement .................................................................................................................... 5
  1.4 Purpose and Research Question ................................................................................................ 6
  1.5 Delimitations ............................................................................................................................ 6
  1.6 Software Requirements ............................................................................................................ 6

2 Method ........................................................................................................................................ 7
  2.1 Design of the Game ................................................................................................................... 7
  2.2 Game Implementation .............................................................................................................. 9
    2.2.1 Networking ....................................................................................................................... 9
    2.2.2 AI ....................................................................................................................................... 9
    2.2.3 Recording Sessions ........................................................................................................... 10
    2.2.4 Feedback .......................................................................................................................... 11
    2.2.5 Scoreboard ....................................................................................................................... 12
  2.3 Experiments ............................................................................................................................... 13
    2.3.1 First Phase of Experiments ............................................................................................... 13
    2.3.2 Second Phase of Experiments .......................................................................................... 13

3 Results .......................................................................................................................................... 14


1 Introduction

Technology has developed greatly during the years, with ever increasing amount of operations that can be executed per second. The development within games, shooter games in particular, has been mostly about graphics rendering, and realistic visual representations of game worlds. The Artificial Intelligence in such games rarely feel alive, and computer controlled units will in almost every case run to its death in stupidity. This has led to an Artificial Intelligence which does not pose a challenge for human players, as it can be easily exploited once the player understands the design of the AI.

1.1 Background

Artificial Intelligence in games has been around since the early beginning of computer game development. It was first mainly used in games to replace human players as opponents in strategic games like chess or checkers, but has since then grown into controlling several different aspects of games. One example is The Director, an Artificial Intelligence developed by Valve Corporation that is designing the game experience while playing the game Left 4 Dead (Left4dead.wikia.com, 2015).

The usage of machine learning with Artificial Intelligence, such as neural networks, has also been around for a long time. An example is the Artificial Intelligence that learned to play checkers developed by Arthur Samuel in the 1950s (Stanford.edu, 2006).

1.1.1 Artificial Intelligence

Artificial Intelligence is a widely researched area in computer science and is most likely the most complex branch of science. The term was first coined by John McCarthy in 1955, himself defining it as “the science and engineering of making intelligent machines” (Stanford.edu, 2007).

1.1.2 Human Brain Project

The Human Brain Project is a large research project established in 2013 that aims to simulate a human brain through computers. If the project succeeds, which is highly unlikely, it will be the first fully working simulation of an entire human brain in existence. The project needs a computer nearly thousand times faster than today’s standard to successfully simulate a human brain, or many computers wired and working together (Humanbrainproject.eu, 2015).
1.1.3 Turing Test

The Turing Test was developed by Alan Turing in 1950 with the purpose to evaluate how well a machine does in imitating a human (Turing, 1950). The test is performed using one human and one machine, both trying to convince a human judge that they are a human. The conversations used in the test are limited to a text-only channel to remove any dependency the machine might have to render text into audio. The machine has passed the test if the judge is unable to distinguish the machine from the human.

1.1.4 Previous Research

Previous research includes John Laird researching human-level artificial intelligence in computer games (Laird, 2002). This research was mainly performed on the game Quake II, which is a 3D first-person shooter developed by id Software in 1997. Their research involves creating a computer controlled unit playing the game as a human would, using their own developed artificial intelligence architecture, called Soar. The AI would be fed with sensory information regarding its environment, similar to what is available to a human, and thus it would also use similar controls to those used by a human. It would use similar tactics as human players tend to use while playing a game of this specific genre. Instead of having already provided for navigational data for the levels, the AI would build its own data by observing and remembering. The AI was also able to react to sounds made by other nearby characters in the game. Experiments with other players showed that less accurate aiming or slower reaction time (around 100 milliseconds) would provide a more natural human-like behavior. During the experiments, qualitative analysis was also performed of the behavior of the players. It was noticed that experienced players attempted to anticipate the actions of their opponents. The anticipation was solved using the same decision making system, but instead utilizing the estimated world views of the opponents. The AI would continue to predict the opponents’ moves until it could find a solution and get to the opponents’ destination before they could.

Another interesting research was made by Darran Jamieson and his teammates when developing a game called Infected (Jamieson, 2015). When they developed the first version of the AI, they quickly noticed that it was predictable and played the same way every time, where the circumstances were the same. They then came to the idea that the AI lacked emotions. In turn, Darran and his teammates started on a path to simulate emotions, using personalities that would give an artificial weight to certain strategies. The personalities which were introduced: reckless, defensive, and explorer. They noticed that this was a massive improvement, as it would add unpredictability to enemy moves and make every game more unique.

The issue that Jamieson still found was that it was not rewarding to beat the AI. He described how the AI viewed the game as a mathematical puzzle to be solved, rather than a game against complex human players. The AI would make moves that were good in the short-term but disastrous in the long-term. The solution for this was for Jamieson to add happiness and fairness
into the system. The AI would be happy or unhappy depending on actions other players performed in the game, and that they would use fairness as a way to avoid attacking players that have recently been attacked. These changes have made the game more personal, and taking certain actions caused noticeable consequences in the game. Jamieson’s conclusion states that a developer needs to decide what kind of AI to have. If enemies are to just mindlessly rush to the player and walk head-first into pits, or to develop something more, to provide a more personal level of interactions with the player.

Jamieson asks the reader to remember three key points (Jamieson, 2015):

- If bots make mistakes, then they feel more human, and less like a "pack of cards".
- If they deliberately make plays based on individual personality (or emotion, like revenge), then unpredictable and exciting gameplay is more likely to emerge.
- If bots try to play cleverly, rather than shortsightedly, they are more likely to make the game fair to our eyes.

1.2 Definitions

Due to the technical nature of this paper, the used technical terms are defined as following:

1.2.1 Game

A game can be defined in many different ways, and may vary depending on personal experiences. The definition used in this thesis will be based on the definition that Ernest Adams used in his book Fundamentals of Game Design (Adams, 2009):

A game is a type of play activity, conducted in the context of a pretended reality, in which the participant(s) try to achieve at least one arbitrary, nontrivial goal by acting in accordance with rules.

In regard to this definition, this thesis paper will be focused on the games developed on and played on a computer. This is not limited to personal computers, but also includes gaming consoles, smartphones, tablets, etc.

1.2.2 Game Engine

A game engine is a collection of software frameworks and tools designed to develop games. Game engines are used by game developers to create games both faster and easier, by using the already developed features and tools provided by the game engine.
1.2.3 **Artificial Intelligence**

In computer science, artificial intelligence has an important place with its studies based on simulating human thinking and behavior by machines (Köse, 2012). In this context, this thesis will cover the specific Artificial Intelligence (AI) used in games, rather than AI in general. In short, the discussed AI implementations do not intend to mimic human thinking. This thesis will instead be focused on implementations that mimic elements of human behavior, with the aspect of performance taken into consideration.

Precisely as Steve Rabin defines game AI in his book Game AI Pro (Rabin, 2014):

> We use the term game AI to describe AI, which is focused on creating the appearance of intelligence, and on creating a particular experience for the viewer, rather than being focused on creating true intelligence as it exists in human beings.

1.2.4 **Bots**

In games, a *bot* is an instance of AI that controls one or more units that share the same virtual brain. Each *bot* in a game can have different properties and personalities. As for players, each *bot* has their own set of goals, most often apart from each other. This does not limit the *bots* from sharing goals, but simply refers to each *bot* being separated from one another.

1.2.5 **Neural Network**

Neural networks are a family of statistical learning algorithms inspired by the brain in particular. Neural networks are commonly used to estimate or approximate functions that are generally unknown. These neural networks are well used within the AI community. As the website WhatIs puts it (WhatIs.com, 2015a):

> In information technology, a neural network is a system of programs and data structures that approximates the operation of the human brain. A neural network usually involves a large number of processors operating in parallel, each with its own small sphere of knowledge and access to data in its local memory. Typically, a neural network is initially "trained" or fed large amounts of data and rules about data relationships (for example, "A grandfather is older than a person's father"). A program can then tell the network how to behave in response to an external stimulus (for example, to input from a computer user who is interacting with the network) or can initiate activity on its own (within the limits of its access to the external world).
1.2.6 Finite-State Machine

A finite-state machine is a state machine with a finite number of states. The state machine can only operate one state at any given time. A transition between states can occur in connection to a triggered event or condition. As the website WhatIs describes a state machine (WhatIs.com, 2015b):

In general, a state machine is any device that stores the status of something at a given time and can operate on input to change the status and/or cause an action or output to take place for any given change. A computer is basically a state machine and each machine instruction is input that changes one or more states and may cause other actions to take place.

To continue with the definition of a finite-state machine. As the website WhatIs puts it (WhatIs.com, 2015b):

A finite state machine is one that has a limited or finite number of possible states.

1.2.7 Remote Procedure Call

A Remote Procedure Call is a way of communicating over network. As the website WhatIs puts it (WhatIs.com, 2015c):

Remote Procedure Call (RPC) is a protocol that one program can use to request a service from a program located in another computer in a network without having to understand network details.

1.3 Problem Statement

The most common choice of implementation of difficulty in AI does not modify the way it thinks, but it rather changes variables in its actions, e.g. reduced or increased precision and/or reaction time. This kind of implementation is transparent and does not help human players to feel the reality of the opposing or assisting units in a game. The players are rarely convinced that the AI is actually intelligent, with the belief that the computer-controlled units only work due to the fact that they have a cheating advantage by having access to otherwise hidden information within the game.

AI could be extremely expensive in terms of processing cycles and memory, so naturally in games the goal is not to recreate the human brain, but instead to mimic specific human behaviors with as little performance impact as possible.
1.4 Purpose and Research Question

The purpose of this thesis is to examine what makes AI seem intelligent for a human player. Computer-controlled opponents in a specific game implementation will be examined in order to study what players view and value. During this research, included in the game implementation there will be an attempted implementation of AI with limited performance requirements and by the players believed to be near equal or equally as intelligent as a human player. The focus of the AI will be of the movement of avatars in a game world.

The research questions for this thesis are:

- How do we design AI in shooting games to seem more human?
- How well would a finite-state machine together with a neural network perform when faking human behavior?

1.5 Delimitations

Due to time limitations of this thesis, there will not be time to evaluate different implementations of AI. There will also not be time to further iterate on the experiments performed. This will result in only discussing results of several tests made on one specific implementation of AI.

1.6 Software Requirements

Since the goal of this thesis is to evaluate how well human behavior of movements can be faked, the human players of the experiments should never be aware of other human players present within the environment of experiments. The solution for this is to allow the human players to participate at any location of their choice using network communication, given they have access to the internet at the given location.

The game also lacks means of communication between the players, since it would require the AI to be sufficient in text or voice communication.
2 Method

The game was developed using the free version 4.6.3f1 of the Unity game engine (Unity3d.com, 2015). The choice of engine was because of the easy management and fast workflow that allows for a much quicker and efficient game development process for a single developer. During the development of the game, version control was used to add safety during development by providing backups among many other generous features.

2.1 Design of the Game

In order to properly experiment with, and test the movements of computer controlled units, a top-down shooter game was selected. The game was developed in 2D to save development time, due to more easily produced graphics and for easier implementations of the math involved in creating such a game. The players in the game are facing other opponents that the player has to eliminate. Each round ends when there is only one survivor left. To make sure that all players were on equal grounds a countdown of three seconds were used before the start of each round, in preparation of the new round. During this countdown, abilities and movements are disabled. A total of ten rounds are played before the game is finished and feedback is prompted for. Each unit would have its own unique color so as not to be easily confused with one another.

The abilities created were teleportation and throwing snowballs. Teleportation was designed to be both a defensive and aggressive ability, where the player could either dodge incoming projectiles or make a sneak-attack by greatly reducing distance to another player. The snowball throwing was designed to mainly be used aggressively in attempts to eliminate opposing players, but could also be used in defensive purposes, such as destroying another player’s snowball.
Figure 1. The game view. Two players dead on the ground while three other players are still alive and fighting. The aim of each player can be seen as green lines and the snowball is the white circle with a trail of white dots behind it. The purple and pink dots are effects after a player teleportation.

Figure 2. Another image showing the game view with three visible players battling each other.
2.2  Game Implementation

A new 2D project was created in the Unity game engine without any additional asset packages and was uploaded to a version control repository. The folder structure was created in an organized way, to separate graphical assets from code. Graphical assets was then imported into the project from other existing projects with permission from the authors of those projects. Textures were set up to be rendered with the correct origin and scale, and particle effects were created for the teleport feature.

Pre-defined objects (called prefabs in Unity game engine) were then created with collision and rendering materials included for walls used in the environment and the avatars of the players. These objects would then be used for easier and faster placement and instantiation of these objects in the game world. The static level the players were going to battle on was then designed using the created pre-defined objects, a texture for the ground and the border walls to prevent the players from exiting the restricted game field.

2.2.1  Networking

Due to Unity providing build-in support for networking, the choice was to use their implementation and work from there. All communication between server and clients were performed through using RPCs. Every unit, both human and computer controlled, synchronized their movements by sending the pressed keys on the keyboard since the computer controlled units had the same interface of interaction as the human players. Every frame, predictions of the unit movements could be made from the keys in which the unit had pressed at the last update. Every second, each unit synchronized its position within the game world to ensure that the units did not drift away during the time predictions that were being used.

2.2.2  AI

The AI was implemented using a finite-state machine combined together with a neural network. The neural network had five input neurons as follows: projectile incoming, enemy close, fire cooldown, teleport cooldown, and estimated enemy fire cooldown. Projectile incoming is a floating-point number between zero and one, with zero being no projectile directed towards the player and one being a projectile headed straight at the player. Enemy close is a floating-point number between zero and one, with zero being no enemy nearby and one being an enemy standing right beside you. Fire cooldown is a normalized floating-point number reflecting how long until the fire ability can be used again. Teleport cooldown is also a normalized floating-point number but instead reflects how long until the teleport ability can be used again. The last input, estimated enemy fire cooldown, is a normalized floating-point number that reflects an estimation of the fire cooldown of the closest enemy.
The neural network operates once every few milliseconds to simulate reaction time, with each computer controlled player having a different reaction time. The neural network operates with the inputs described above, with the results being one floating-point for each state in the state machine. The state with the highest result value is transitioned to, at the same frame as the calculation.

The neural network was trained from a manually crafted text document with samples of different input values and the expected state from those inputs. The manually crafted text document had 58 lines with samples for each state and took less than one second to train. The text document used for training is provided in Appendix C. The error threshold of the learning algorithm was kept at 7 due to the nature of the manually crafted samples. Attempting a lower error threshold ended in a never-ending loop when training the AI. The learning rate was set to one tenth and the learning algorithm used was a one-layer perceptron (Rojas, 1996).

The neural network was implemented by using AForge.NET framework (AForge.NET, 2015). The libraries that are used from the framework are the following: AForge Core, AForge Fuzzy, AForge Genetic, AForge Math, and AForge Neuro.

The states used in this implementation are as follows: dodge, fleeing, search, hunt, and firing. The dodge state simply forces the unit to move out of the way for any incoming projectile. The fleeing state forces the unit to move away from the closest opposing unit. The search state moves the unit around the level in attempt to find opposing units. The hunt state makes the unit follow and hunt down the closest opposing unit, in attempt to eliminate them. The last state, firing, inherits the same functionality as the hunt state but also fires shots when the aim is pointing close enough to the targeted unit.

As mentioned earlier, the computer controlled units used the same interface of interaction as the human players did. When the AI wanted to move the controlled unit or fire a shot, the AI ordered keyboard presses that would be handled the same way as for regular keyboard input from players.

To navigate, the AI was using the A* search algorithm (World of Computing, 2015). At the start of the game, all nodes for potential areas of walking are generated based on the collision data of the level. Every time the AI needs to move somewhere else, it is using the search algorithm to avoid moving through walls and other obstacles.

### 2.2.3 Recording Sessions

To enable the collection of more data from each session of an experiment, a feature of recording sessions was implemented. The sessions were recorded by saving data communication the server was broadcasting to all connected clients (human players). The data recorded included timestamps to be able to execute the session in replay at the same point in time.
The people during the later experiment were handed these recorded sessions that they could view as many times as they wished. Once the people watching these recorded sessions were ready they were provided with a template for feedback where they could document events and better describe how and what tipped them off in the analysis of the AI controlled units.

2.2.4 Feedback

After ten rounds of an active game session, a feedback screen appeared, prompting the user to give feedback of the played session. The player would get to see a list of all players in the session that would be distinguished by their color. The player was prompted to click on each player that they suspected was computer controlled. Once the player clicked on a picture of a player, they would be sent to a new screen where they could describe the reason for why they selected that player. The feedback was sent to the server for collection and the player was sent back to the list of players to be able to select more than one player. As the player was finished there was a button to simply submit the feedback and notify the server that they had indeed finished their feedback.

The server would see a scoreboard with the total scores collected from the ten played rounds. The measured scores were as follows: kills, deaths and survives. There would also be a voting table visible for real time visualizations of who players voted for. In addition, a log was also provided, to be able to see the events of people voting and to be notified when all players had finished their feedback. The feedback received to the server would also be saved to text documents on the hard drive to prevent loss of data.

Once the players had finished their feedback, they were allowed to leave the game, as this would no longer cause harm to the data collection.

![Voting](image)

**Figure 3.** *The overview of all player avatars in the game, with the players own avatar being grayed out. Clicking on an icon of an avatar would bring up a new screen for voting on that specific player (shown in Figure 4).*
Figure 4. *The voting screen where one could describe the reason for voting on a specific player.*

Figure 5. *The scoreboard and voting table seen only by the server during the feedback phase.*

### 2.2.5 Scoreboard

A scoreboard was implemented on the server side that could only be viewed by the host of the session. The scores measured were as follows: *kill count*, *death count*, and *survive count*. The *kill count* was increased by one for every player that got killed by a snowball thrown by the player. The *death count* was increased by one for every time a player died. The last score, *survive count*, was increased by one for each time the player was the last survivor on a round.
2.3 Experiments

For each experiment performed, people were recruited using social platforms such as Facebook and Skype. There were no specific rule of selecting the human players for these experiments, as every person interested got recruited.

The experiments were performed on Windows computers where every person involved were playing from their home. Before the experiments, each player would download the game files from the online upload service of Google Drive. The players would also be handed a description of what the goal of the experiment was, and how the game was played.

The data collected from both performed experiments were always sent to a server that would serialize the data into a human readable format in text files, and save the text files on the local hard drive of the server, as previously mentioned.

2.3.1 First Phase of Experiments

The first phase of experiments involved gathering as many human players as possible to play through two game sessions. The first game session included four human players with two active AI controlled units. The second game session included three human players with four active AI controlled units. Every game session started with waiting for every player to connect to the game and announce that they were ready. The players could not see any information regarding the other players, neither human nor AI. As every player had announced that they were ready to the host of the game session, the first round was started. The host of the game session could spectate everything within the game session live with controls to move the camera freely in any direction.

2.3.2 Second Phase of Experiments

The second phase of the experiments involved gathering humans to view the recorded game sessions. These viewers are referred to as spectators. They would do so by downloading files containing the recorded data of the game sessions, and view them from within the same game client that was used when playing the game sessions. There were no restrictions for viewing recorded sessions, as the spectators could view any replay as many times as they deemed necessary to get a clear picture for the feedback. During this phase all spectators could tell human and AI players apart in order to allow more detailed feedback of each specific player. The feedback requested from the viewer was general feedback of the sessions as well as specific feedback for certain events and player actions. The feedback was restricted to only mentioning details about AI or human behavior.
3 Results

The results from the game sessions were compiled into lists with different categories. Every option mentioned in the feedback from players was added to the list, where similar options were merged into one option, for a better overview. The data used to gather the following results is provided in Appendix A for the combined feedbacks from the players of the two game sessions and Appendix B for the combined feedbacks from the spectators.

3.1 Placement of Votes

![Pie chart showing the results of vote placement.]

Figure 6. The results for placement of votes at the feedback screen from both played game sessions.

Figure 6 shows the share of votes that were correctly placed on AI controlled players, the votes that were incorrectly placed on human players, and finally, the votes that were lacking for all players to place at least one vote for every AI controlled player in the game.

Gathered from two game sessions, with seven players in total, the same amount of votes were correctly placed as incorrectly placed.
Figure 7. The statistics of placed votes relative to their position on the scoreboard.

Figure 7 shows the statistics of placed votes relative to their position on the scoreboard, sorted by kill and death counts. This data is provided to show the connection between how well the player was performing and the amount of votes placed on that player. It is not specified whether the player was human or AI controlled.
3.2 Reason for Voting on AI

Figure 8. The statistics of the reasons why players voted on AI controlled players.

Figure 8 shows statistics of reasons used for why votes were placed on AI controlled players. Multiple reasons could be used for a single vote. While most of it is spread out, perfect aiming and fast reaction accounts for the most common reasons for believing a player was AI controlled. This data was collected for each vote placed by a human player.
3.3 Reason for Voting on Humans

Figure 9. The statistics of the reasons why players voted on other human players.

Figure 9 shows the statistics of reasons used for why votes were placed incorrectly on human players. Multiple reasons could be used for a single vote. In this case perfect aiming accounts for the biggest reason overall for falsely accusing a human player for being AI controlled. This data was collected for each vote placed by a human player.
3.4 Spectator Feedback

Figure 10. The statistics of common characteristics for the AI players from the viewpoint of a spectator.

Figure 10 outlines the common characteristics for the AI players that were summed up from the feedback from six spectators that viewed the recorded sessions. Spectators were not limited to only one characteristic in their feedback. All spectators reported that the AI players were up close and personal and rarely shot from a larger distance. A majority of spectators also reported that the AI players had good timing on their abilities, specifically using teleport to dodge incoming projectiles.

Spectators also reported that the AI players seemed more experienced with the game. The AI players knew the cool downs of abilities well, and had strategies they used even from the start of the game. For most new human players it took a few rounds before they got more comfortable with the game mechanics.
Figure 11. The statistics of common characteristics for the human players from the viewpoint of a spectator.

Figure 11 outlines the common characteristics for the human players also summed up from the feedback from the six spectators that viewed the recorded sessions. Spectators were not limited to only one characteristic in their feedback. Two things every spectator reported in their feedback was that most human players played very carefully and were more afraid of dying than the AI players. Another thing every spectator reported was the fact that all players had different and unique play styles, meanwhile the AI players appeared to have the same play style. A great majority also thought that the human players were looking around much more with their aim, rather than looking where they were headed or always towards another player like the AI players tended to. Another characteristic specific to human players was that they could fire blindly into areas they did not see, in hopes of hitting something.
4 Analysis

When analyzing the data collected during the experiments, earlier provided figures (Figure 6-11) and the recorded game sessions provide enough information to analyze the experiences of the human players and compare them to events in the game sessions.

4.1 Voting

Exactly half of the votes were misplaced on human players and the biggest reason for this was because they had very good aiming skills. Taking into account that some of the human players were generally skilled players and some not as skilled, one assumption would be that the not so skilled players that did not adapt well into the game voted on skilled players because they had much better aim. No human player knew exactly how many AI controlled players or human players that were in the game, they only knew the total amount of players in the game. Another assumption would be that all players were sure that AI players would be in the tests, and therefore had to vote for at least one player.

Almost half of the expected votes were lacking, which can be viewed as the human players were not sure about some players, or they simply believed they were human. As there was no actual feedback gathered for why a player did not vote for a specific player, there is unfortunately no way to tell why they did not vote for that specific player. One thing to take into account is that during the ten rounds of a game session, the player might not have seen all other players in the game, or seen a specific player for such a short time to not get an impression of them.

Only half of the vote s were correctly placed on AI controlled players, as mentioned earlier. Looking at Figure 7 again we can see that a larger amount of the votes were placed on the best scoring player in the games on average, but at the same time it is shown that another equally large amount of votes were placed on the fourth best scoring player in the games on average. This means that not all players believe the best scoring player is automatically an AI, even though in both game sessions played an AI was the best scoring player. Having in mind that no players had played the game before, and only had a short text to read for instructions and what to expect from the game. An assumption could be made that the votes could have slightly shifted to target more AI players than human players.

The players that got used to the game quicker were also the players that voted with most accuracy on the AI controlled players. Meanwhile the players that had a harder time with the game, making mistakes and shooting in the wrong direction, had a larger misplaced amount of votes on actual players instead.
4.2 Voting Reasons

The biggest reason behind voting on AI players was that they believed they had very fast reaction, able to dodge incoming projectiles easily. This made them hard to hit without being close to them. Most players reported that they very often teleported away from projectiles, rather than walking out of their path. Meanwhile a few players also grasped the idea of teleporting to dodge incoming projectiles, their AI opponents did it more often. In a case where the players would have the chance to practice playing the game more to get used to it, it could be assumed that humans would use teleportation as a defensive ability to a greater extent.

The second biggest reason behind voting on AI players, and the biggest reason behind voting on human players, are identical to each other. The human players reasoned while voting that the players they voted for were having a perfect aiming ability, always pointing straight at another player. From this, it could be assumed that human players often expect AI players to have a perfect aiming ability, to never miss its target. Therefore even human players with good or perfect aiming ability could be falsely accused of being AI controlled.

4.3 Spectator Feedback

Since the spectators had more time to watch the replays of the played game sessions, they had a clearer picture of the game sessions than the players had. They also knew which players was human or AI in advance, to allow for the more detailed feedback about what their specific characteristics are and what tells them apart.

One of the most common characteristic for the AI players according to the spectators was the faster reaction, which was shared with the feedback from the players. The other equally common characteristic which was surprisingly never mentioned in the feedback from the players, was that all AI players seemed to always get up close and personal, not being afraid of dying. This correlates to one of the most common characteristic for human players, which were being more careful than AI players.

Another thing that the spectators noticed, something that the players reported on other humans instead, was not to waste resources. Spectators noticed how AI players saved their abilities like throwing snowballs until they had a player to throw them at, rather than the usual player approach of blindly throwing in hope of hitting something. What made players vote on other human players for this very same behavior is uncertain due to lack of a more detailed reasoning behind the votes. One assumption could be that one or a few players used the same tactic as the AI players, to save abilities for a bigger certainty and safety. Spectators might have failed to report this for human players as AI players used this tactic to a much greater extent.
5 Reflections

Due to time restraints, many sacrifices had to be made during research and development of this project. Different implementations of AI and different setup of parameters used by the neural network would provide for a comparison of data in-between. Where one or multiple winners could be selected and analyses into why players preferred a specific implementation or setup could be made for a deeper understanding of what players view and value in AI.

5.1 AI

Every AI controlled player had the same neural network (in other words the same personality). The differences between the AI controlled players were different reaction and response times. This made the AI controlled players slightly different, but initially had the exact same responses to the same input, making them all behave in the same way. In the feedback from both the players and the spectators of the game sessions, this was one of the common characteristics for AI players they found.

More sets of data to train the AI with personality in mind could make for larger differences between the AI players. A personality could include cowardice and aggressiveness that would be affecting their judgement.

Reflecting over at the previous research performed by Darran Jamieson and his team, they noticed a huge dynamic impact when introducing personalities and emotions into their game. Assumptions could be made that this would also be the case for most AI implementations in games, no matter which genre.

5.1.1 Based on Expert Players

The AI was based on how an expert player would play the game, to optimize the levels of the challenges for the players, since every AI had the same neural network, and thus reacted the same way. Every AI was essentially an expert player with added timers for limited reaction time. In every game session played, there were at least one human player significantly less skilled. These players did not have a matching AI for their level. Even though some AIs were really slow at making decisions, they still used the same tactics as an expert player. A tactic could be to wait for the other players to use their abilities and then strike to prevent the other players to dodge.

Also because of time restraints, every AI player executed their actions in the same way. Making several implementations of the states in the state machine would give AI players different ways to flee, dodge, or attack.
5.1.2 Aiming

One of the most common characteristic that both players and spectators found was the aiming ability that the AI players had. Most of the players and spectators thought their aiming was too perfect, and they never made mistakes. A lot of focus went down into the aiming, like simulating mouse movements, but it was simply not enough. As mentioned earlier, the AI was modelled after expert players. The AI players did not feel stress, like being surrounded by a crowd of other players and not knowing where to go. The AI players always knew what to do, there was never any uncertainty in their plans. If there was a player close enough to the AI player, it would focus on that player until that player died, or if there were a more imminent threat nearby.

In a case where for example stress was taken into account, the player might get into a state where the aiming is not so accurate for a short period of time. The ability to handle stress might be varying depending on the AI personality.

For aiming, the AI players did not have the ability to approximate where a player was running and fire projectiles in that direction. Instead they always tried to aim directly at the target, which could also be a reason why so many believed they had perfect aim. Meanwhile in reality their aiming was not perfect at all, with all things considered. Human players better used this method of approximation when firing snowballs, making it possible for other players to believe that their aiming was not as good as it actually was.

5.1.3 Looking Around

Another common characteristic that both players and spectators agreed upon was the fact that humans looked around the world more, and would have more movement in their aim when running around looking for other players. As well as with aiming, a lot of time was spent trying to make this believable for the AI players. The implementation included making the AI player move around the mouse simulating where a human player might look when running around looking for other players. Apparently this was not good enough to be believable. The implementation of this feature was based on a randomized look around, using the center of the level as a reference point to make sure that the AI would not stare into a border wall, where nothing lies behind. An idea to make this more believable might be to change the point the AI is looking at more often to make it look smoother. Another thing would be if the AI would look towards where other players’ laser aims come from, or where it last saw projectiles coming from, and take more things into consideration rather than to select a look-at point by chance.
5.1.4 Limited to One Target

The implementation of the AI was limited to only consider one target at any given time. The target might change, but it is always only one target. The reason for this was because adding more potential targets would make the neural network exponentially more complex for every target.

5.1.5 Neural Network and Finite-State Machine

Using a neural network that controls which state a finite-state machine should transition to works great and has saved a lot of development time. Meanwhile there are a lot of other options into creating a virtual mind that mimics human behavior, a neural network combined with a finite-state machine is a good candidate. The implementation used for this thesis can further be improved to become even better, but has still proved to be good enough for the time spent with it.

The gains of using a neural network for AI is that a lot of complexity can be gained for very little effort and development time. Since neural networks are simulations of how a brain works, with less processing power. Although a true simulation of a brain is believed to be possible with enough processing power and memory. The human brain project is an example of such a belief, but it has not yet been tested to the extent that it can be written off as true.

Finite-state machines has been a very widely spread system often used in games and other computer software. It is very simple to implement and also very easy to understand with visual representations. Combining the neural network with a finite-state machine can be considered to require relatively little time to implement, and a little more time to tune. The results are however extra-ordinarily good for being such an easy implementation. Considering the implementation performed in this project was rushed under a time restraint, a much better implementation is a definite possibility.

5.2 Feedback

The feedback screen after a played game session allowed all human players to vote for the suspected AI controlled players in the game and give a reason into why they gave the vote. The improvement to this would be to allow the players to also give reasons into why they did not give a specific player a vote, as this would allow for a much greater detail on the data returned from the experiments. With the setup used for these experiments, there was no way to tell why votes were left out and never used. By having this information in the experimentation data, a filter could have been used to only count votes the player was sure about.

Feedback for spectators could have been done in multiple ways, different from this setup. A desired method of feedback for spectators were to use the same method as the human players used inside the game after they had played a game session. This method could have been used
additionally to the feedback method used in this setup. That way it would be possible to gather much more data, as a spectator requires less time and scheduling than a player would. When hosting a game session, human players had to be recruited for the experimentation. Preferably having multiple human players in the same session. During the entire time the host had to be active in the experiment and guide the players through it. When asking a spectator to spectate the played sessions, they were given a package of instructions, data, and executable files. During no point in time other than the actual handover of the package and the feedback was the host actually present. This meant allowing the host to send out packages to a much bigger group of humans with less time required to take care of it.

5.3 Unity

Using Unity game engine for implementation of the game was a good decision overall, since the entire game could be implemented in a very short time. The gameplay was developed in only three days.

The current implementation of the networking support within the Unity game engine in version 4.6.3f1 caused issues because of lacking features and functionality. Much time was spent on getting vital functionality to properly function as desired. Time that would be better served on the AI implementation and training. This could have been less time consuming if a third-party library for networking in Unity would have been used instead.
6 Conclusion

The conclusion from this research project is that faking the human behavior is very much possible without having to create a virtual human brain. But much more research is necessary in this field in order to find the perfect solution for this.

The answer to the first research question, “How do we design AI in shooting games to seem more human?”, is hard to answer as it depends on the specific game in question. However, one thing learned from this research project was that players expect AI players to be perfect and never make mistakes. While human players, expert or novice, makes mistakes. Different games has different goals with the AI. Speaking in context of this thesis, with believable human-like movement behaviors, personalities and simulated emotions can be assumed to make an AI even more believable. The key here would be to design AI players as they were humans, define their play style and preferable choices. Items to consider on their personality might be how self-secure they are, their tendency to search out other opponents and their standpoints on risks.

Since humans are very driven by emotions like for example anger, jealousy and fright, implementing emotions to your bots will also be assumed to increase the realism in faking their human behaviors. Even in a game as small as the one implemented as a part of this research project, emotion like fear of dying from an opponent that the AI knows has been dangerous before, will change the way it is playing. This can make the AI seem more dynamic, even though it can be as simple as adding one more input into the neural network.

Regarding the second research question: “How well would a finite-state machine together with a neural network perform when faking human behavior?” A neural network that controls transitions between states in a finite-state machine works really well, and a definite candidate for faking human behavior. More samples for training data, different training data for each personality for the neural network, and more focus on individuality along with improvements on specific actions will make it even more convincing in faking human behavior.
References

Literature


Internet


Appendices

Appendix A

Total Game Sessions: 2
Total Players: 7
Total AIs: 6

Voting Total

- Correct Vote - 5
- Misplaced Vote - 5
- Missed Vote - 9

Voting - AI

- Perfect aim - 3
- Fast reaction - 4
- Fast movement - 1
- Not looking around - 2
- Predictable - 1

Voting - Humans

- Perfect aim - 4
- Fast reaction - 2
- Perfect execution - 1
- Didn't waste abilities – 2
Appendix B

Total Spectators: 6

AI characteristics

- Good timing - 5
- Excellect dodge skills - 5
- Up close and personal - 6
- Doesn't waste abilities - 3
- Faster reaction - 6
- Uses telport more often - 4
- Seems more experienced - 2

Human characteristics

- Firing blindly - 3
- More careful - 6
- Different playstyles - 6
- Does not teleport often - 4
- Looks around much more – 5
Appendix C

1;0;0;0;0
1;1;0;0;0
1;0;1;0;0
1;1;1;0;0
1;0;0;1;0
1;1;0;1;0
1;1;1;1;0
1;0;0;0;1
1;1;0;0;1
1;1;0;1;0
1;1;1;0;1
1;1;1;1;1

0;1;1;1;0;1
0;1;0.75;1;0;1
0;1;0.5;1;0;1
0;1;0.25;1;0;1
0;1;1;0.75;0;1
0;1;0.75;0.75;0;1
0;1;0.5;0.75;0;1
0;1;0.25;0.75;0;1
0;1;1;0.5;0;1
0;1;0.75;0.5;0;1
0;1;0.5;0.5;0;1
0;1;0.25;0.5;0;1
0;1;1;0.25;0;1
0;1;0.75;0.25;0;1