Human-Like AI in Real-Time Strategy Games

Duncan John Bunting

BSc (Hons) Computer Games Technology

2017

Institute of Arts, Media and Computer Games
University of Abertay Dundee

Table of Contents

List of Figures

List of Tables

List of Equations

Acknowledgements

I would like to take this opportunity to acknowledge all those who helped me through my time at University and for this project. Friends, family and strangers; I do not think I would be here today if not for the support of everyone during the past 4 years. Thanks to Conor Paterson, Ciaran Walker, Shane Ellis, Craig Jeffrey, Rory McLean and John Bruce for accompanying me on this journey and keeping me reasonably sane. Thanks to all the team members of Cybernaut Games and 8Click Studios. Thank you to all those who sat through the incredibly long questionnaire and testing session and provided the feedback for this project and the game in general - I am sorry I tricked you!

Special thank you to my lecturers, supervisors, mentors and tutors at Abertay University, particularly Iain Donald, David King and Grant Clarke who were supportive throughout my learning and helped me establish my interests and goals in game development and gave me the confidence I desperately needed. As well as the continued support that everyone provides as I take the scary transition into the games industry.

A special thank you to my Step Mum who put up with proof reading all my essays despite not knowing what they talked about most of the time! My Dad and Grandpa for supporting me through University; words cannot express how thankful I am to have such a wonderful, supportive and understanding family. I hope I have and continue to fulfill my families expectations and make them proud.

Finally I'd like to give a special thanks and credit to my little brother Caelan Bunting; who assisted a great deal in the bug hunting and creation of maps for the game. Without him I would have definitely lost my mind! Hopefully I have not tainted his interest in game development and he continues to explore that path and perhaps one day I will read his dissertation.

Abstract

Context

Artificial Intelligence (AI) in video games has been ignored for a long time in the rapid advancement in technology and game development; however, in recent years the topic of AI has resurfaced presenting a large talking point in the games industry and other professions. One of the reasons for this rise in discussion over AI in the games industry is the demand for a more human-like AI to bring the singleplayer experience closer to what is currently only obtainable in multiplayer modes.

Aim

This project set out to research and develop an AI that can be considered human-like through the use of different AI techniques and implementation of various human qualities in a real-time strategy game environment and investigate whether these additions reflect positively on the player experience.

Method

To achieve this target two AI techniques, Rule Base and a customized version of Fuzzy Logic, were implemented into a real-time strategy game environment; Little Planets. Each AI technique was then adapted to include human-like qualities such as reaction times and development of trust creating two additional AIs, one for each technique. Finally the Fuzzy Logic system was adapted to output probability rather than fuzzy sets in an attempt to have an AI mimic human error with the ultimate goal to create an AI that can pass as a human player.

Results

AIs were put through a series of tests performed by volunteer testers which evaluated the realism, enjoyment and challenge. This study found a steady increase in realism through the AIs with the probability system being rated the most human-like. As a final test to evaluate the ability of the AI to pass a human, testers were lead to believe they were about to conduct the Turing Test. The test asked the tester to identify which of the 5 AI they believed to be a human player after playing against them over three matches. All testers believed they were actually facing a human player and voted the probability fuzzy logic system to be the human player; again followed by the simple rule base system with testers noting they selected it because it was the best.

Conclusion

The results of this study found that probability fuzzy logic or some other implementation of a probability technique may be the solution to creating a more human-like AI in games. This study's results also speak volumes to the current state of AI wherein that an AI was believed to be human by some testers simply by being the best. In addition the study found that the inclusion of human-like AI does increase player enjoyment without affecting the challenge the AI poses negatively - in some cases the challenge increased. Therefore this study can conclude that probability AI systems are the most human-like out of the tested systems and the inclusion of human-like qualities does reflect positively overall on the player experience. Proving that it is worthwhile for developers to invest more in improving the AI in their games.

Abbreviations, Symbols and Notation

AI	-	Artificial Intelligence
ANN	-	Artificial Neural Network
APM	-	Actions Per Minute
APS	-	Actions Per Second
RT	-	Reaction Time
RTS	-	Real-Time Strategy
SFL	-	Standard Fuzzy Logic
4X	-	eXplore, eXpand, eXploit, and eXterminate (Strategy Game Genre).

1 Introduction

1.1 Project Concept

Artificial Intelligence (AI) in the games industry is considerably different to what academics would consider to be AI. Academics are process oriented while games AI is mostly results oriented, the goal for academics is to solve the algorithm and how the problem is solved. Games AI programmers and designers generally do not care what is controlling a character or the realistic accuracy of the decision making, so long as the decision makes sense and does not break immersion. Neither side of the argument is “correct” but each with their own goals and motivations; this unfortunately has lead to AI in games often being forgotten about or pushed aside to invest more time in other areas of development. For the most part games AI has never had the need to be robust and often not even remotely considered intelligent so long as it serves its purpose. This has left very basic AI being implemented into very complex games leaving plentiful room for improvement.

This project seeked to somewhat merge this line between academic AI and games AI while investigating what qualities separate human and AI players and then implement a solution in a self-made real-time strategy game environment. The game environment is based off the same style of game used for the Google AI Challenge 2010: Planet Wars (Melis, G. 2010) which was inspired by the rebirth of this type of game by Galcon (Hassey, P. 2008). Since Galcon’s release many clones and adaptations of the game have been made while keeping the same basic rule-based. These games, although extremely popular have a very clear falloff in popularity once the user has figured out the AIs ruleset. One of the most loved yet least used features in a RTS game, especially those with a long average game-time is multiplayer. Human players love playing against other human players not only for the variety involved with playing against various human players from around the world but the extra set of “mind games” that can, for now, only be achieved in a human vs human multiplayer experience; hence why we need AI that can ‘replicate’ a human player to give the same game experience.

The results of this research into human-like AI in real-time strategy games will hopefully inspire and promote the refocus of attention to the quality of AI in video games and merging the bridge between the singleplayer and multiplayer experience.

1.2 Artificial Intelligence Techniques

Rule based systems are often the go-to for games of this type as the core rules behind the game are often very simple and easily defined; however, rule based systems are the main system that ends up being exploited by players due to the static rules and therefore being easy to exploit. Therefore there is a need for a system that presents a form of unpredictability while still being reliable enough to win the game.

Fuzzy Logic presents itself as an effective solution due to the variation in output compared to a boolean rule-based system; however, although the output of a Fuzzy Logic System is determined by the degree of truth from more than one rule this decision making can still be reduced to crisp binary decisions. Fuzzy Logic is deterministic meaning it can be predictable if the rules are discovered; unlike a non-deterministic system which would produce unexpected results. This ends up giving an AI that is more complicated but will, for the most part, produce similar results as a rule-based AI. To add unpredictability while still making logical decisions this project will not only implement several rule-based and matching Standard Fuzzy Logic systems but also a Probability Fuzzy Logic system which should increase unpredictability and provide a margin of error from making the “correct” decision.

1.3 Research Question

Which AI techniques are more effective at simulating a human-player’s actions in a Real-Time Strategy game and how does having an AI with human-like qualities affect the player experience?

2 Literature Review

2.1 AI in Games

Artificial Intelligence (AI) has played such a critical role in games since its inception. The first example of AI in a computer game came in the form of the mathematical strategy game Nim built on one of the earliest known computers Nimrod (Ferranti 1951) shown in Figure 1 which was custom built to play a game where players took turns removing at least one object from one of a number of heaps, a player could remove any number of objects from the same heap on their turn with the overall goal of the game to avoid being the person to remove the last object. The AI was a great success able to repeatedly win matches against even the most highly skilled players (Grant, Lardner 1952, p. 18).

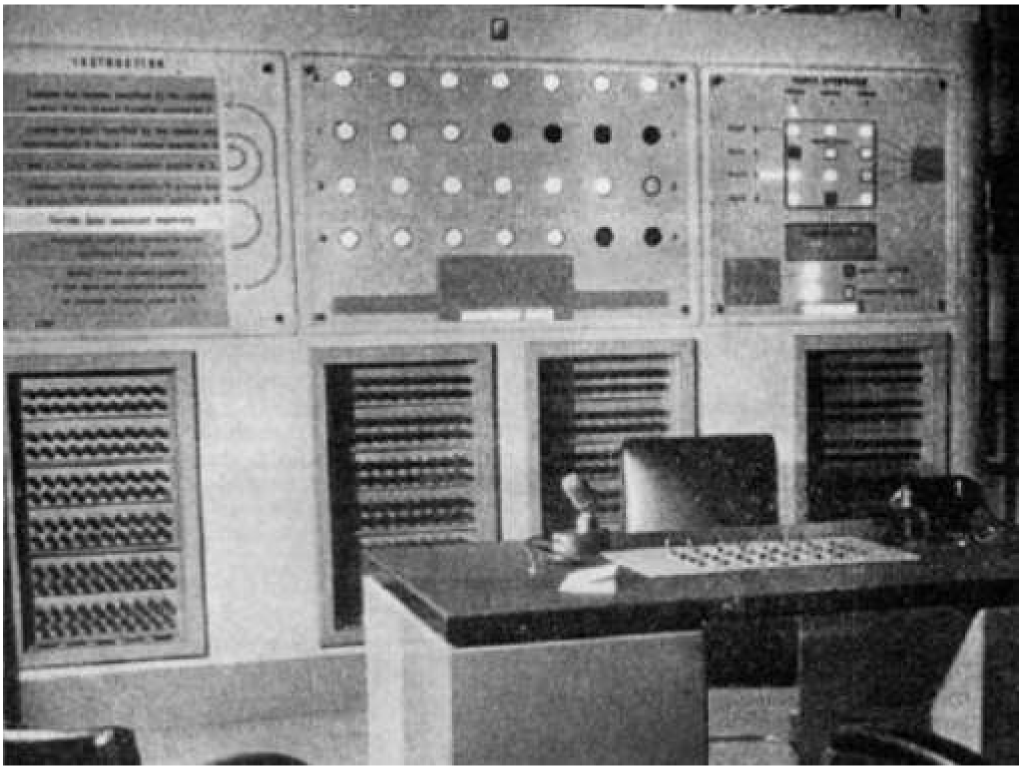

Figure 1 - *Nimrod.* (Electronic Engineering 1951)

As computers became more common the games the first games by Taito Corporation, Speed Race (Taito Corporation 1974) and Gun Fight (Taito Corporation 1975) , emerged where players would face off against artificial intelligence; although the AI was very simple this was great feat in computer game technology allowing the everyday person to experience artificial intelligence for the first time commercially. It was not until the 1990s with the creation of the real-time strategy (RTS) game genre and the revolutionary Dune II (Westwood Studios 1992), shown in Figure 2, being credited to the design of RTS games to this day. It was here that limits of AI truly began to be pushed by taxing the AI with an abundance of tasks. It was around this time that AI had finally pushed the hardware at the time to its limits.

Figure 2 - *Dune II Screenshot.* (Kosta 1992)

Unfortunately since then, while hardware and technology has seen rapid advancement, AI has been left behind such as Anno 2070 (Blue Byte 2011), an insanely complex game with countless features and beautiful aesthetics but an AI that has been labeled "pathetic" (Anon 2013). Games companies believe that for the most part good AI does not sell the game; stating, a good AI is valued and the game may be praised for it but most customers will not buy a game if you simply say the AI is more realistic, the key selling point is to visualize this (Hruska, 2015). This is a big problem for many game genres such as shooters, adventure and role-playing games where visuals are very important and in regards to investing more time into bettering the AI and implementing the visuals for those improvements: most games studios would rather spend that man-power on other areas of the game that sell the game better. Thankfully as the general populace become more aware of artificial intelligence the need to evolve and expand on AI research and implementation into games grows ever-stronger. In response to this resurgence in popularity, it seems game studios are becoming more brave in the assignment of resources to the development of more complex, immersive and AI that can learn. Recent examples include successful Alien: Isolation (Creative Assembly 2014) and ambitious upcoming game Hello Neighbor (tinyBuild 2017) where the enemy AI learns and adapts to the decisions you make and the situations you put yourself and the enemy into; allowing the enemy to counter strategies and forcing the player to try new ones to progress. Not only do both of these games include a more intelligent AI than most games they used the AI as a marketing selling point for the game proving that AI can sell a game. Unfortunately the AI techniques used in these games are not officially publically documented as of writing this; however, several speculate that a pattern recognition system is used.

2.2 Real-Time Strategy Games

Commonly, a Real-Time Strategy (RTS) involves players having one or many bases, units that spawn in these bases and the common goal to overpower and destroy or take over the other players bases to win the game. RTS games have always been a touchy area when it comes to AI as the genre constantly seeks to push AI and technology to its limits.

Figure 3 - *Stellaris Screenshot.* (Daniel Starkey 2016)

From the release of Dune II (Westwood Studios 1992) pushing RTS into mainstream gaming to Stellaris (Paradox Interactive 2016), shown in Figure 3, which has just recently re-popularized the genre. The biggest difficulty with creating RTS games is the sheer complexity (Jensen K. 2016) that strategy games bring to the table. Balancing becomes a constant task and presents the problem of how do you balance an RTS which is a near sandbox of decision making and requires complex long term decision making - Not only do you have to have a whole range of information to govern for which decisions are to be made but also how do you tests such a system. In a presentation at GDC 17 Mehrnaz Amanat Bari, a programmer at Paradox Interactive who worked on Stellaris explained (Bari, M. A 2017) that the Stellaris team had to simply play the game a lot to properly document and assess the AIs behavior to ensure it was not doing something that it should not. This is considered to be the core reason why there are so few 4X games and even fewer successful ones as it requires a lot more testing than other games especially when games as large and complex as Stellaris can have games last 50+hours on a medium sized map.

2.3 AI in Real-Time Strategy Games

One of the genres that is renowned for pushing AI to its limits is real-time strategy (RTS) game AI where there are an abundance of decisions being made by AIs from micromanaging units and resources to long term planning and strategy. Strategy games are also unique for not needing such a large focus on visuals with the player base most content with text and small indicators to express the AIs behaviour meaning that developers can work on complex systems and have the AI perform the majority of its tasks behind the scenes as the game plays. Long running game series such as Total War (Creative Assembly 2000) show a clear improvement with every generation of their game released. Both games implemented a version of human-like opinions or emotions towards others; for example, in Total War: ROME II (Creative Assembly 2014) if a player were to rapidly expand their territory they are labelled around the world as “Expansionists” making other players more wary.

Stellaris (Paradox Interactive 2016) has successfully re-popularised the 4X game genre and content focused heavily on the AI and its decision making uses a data driven design which bases actions and attitude towards other factions (in-game groups controlled by human or AI players - this term will be used throughout the remainder of the dissertation) off the beliefs, ethics and policies of the factions selected in the game, essentially giving AI personalities.

Figure 4 - ***Alpha Ethics System in Stellaris.*** **(Super-Soviet 2015)**

In Stellaris, ethics define the core believes of the faction within the game. Ethics are limited as a faction cannot be a militarist and pacifist at the same time, as seen in Figure 4. An example of this being applied to the AI decision making would be when encountering a new faction, a militarist may tell the faction to stay out their way but a pacifist would welcome the into their society. Down the line these choices also affect the in-game policies that you govern for your civilization, for example a Fanatic Xenophile faction will become extremely upset and possibly rebel if their leader were to enforce purging or enslavement of aliens. Overall how Stellaris utilities data driven design for faction personalities is an excellent way to give a variety of AI behaving differently; however, this design can still be broken down to the same level as a rule-based system making it easily predictable once figured out. The next step for AI in games is to apply a degree of unpredictability which would bring AI so much closer to behaving like a human.

2.4 The Multiplayer Experience

The debate of what is better, multiplayer or singleplayer has been around since the dawn of online gaming and for the most part singleplayer and multiplayer games are vastly different. For strategy games however, the multiplayer mode is exactly the same as singleplayer, except with the AI being replaced by human players. The inclusion of multiplayer modes in strategy games has helped outline how different an AI is to human players even on the most advanced strategy games out there.

Multiplayer adds a whole new layer to any game and this is especially true in strategy games where human strategy tactics are almost limitless. In games like the Civilization game series (Firaxis Games 1991), mind-games between players are at its height. This layer of gameplay, which for the most part is played in the player's head rather than in the game requires immense decision making and long term planning which is simply not something which any AI in modern strategy games is capable. In Civilization V (Firaxis Games 2010) players can form alliances “hidden” alliances or pretend to place a certain way only to have it turn out it was only to trick another player. It is believed that multiplayer strategy games will always provide this unique experience unobtainable from the singleplayer variant giving multiplayer a highly sought after appeal.

However, in almost any online strategy game, developers face the same complaint: games last too long. From the pure amount of power required to process all the information a strategy game has to process over multiple clients of varying computer specifications to the simple fact that humans take longer to decide what to do than computers coupled with the limitation that all human players must be present. “I choose to play at my pace, on my terms, the cast of the game does not huff and grumble” (Walker 2011).

2.5 Humans vs AI

A lot of things separate human players from AI; one of the most significant being reaction and thinking times, this can be a very hard thing to judge but thanks to online tools such as HumanBenchmark.com (Human Benchmark 2017) we have a better idea now of how long it takes humans to react to certain things. In addition the StarCraft video game series (Blizzard Entertainment 1998) is renowned for the investigation done into the importance of reaction time, or how they measure it: Actions Per Minute (APM). Professional level Starcraft players easily pass 200 APM while casual players average around 50 (Starcraft Wiki 2012). These APM values are specific to Starcraft a game that has players perform a huge amount of tasks to be micromanaged in real-time in a fast paced player versus player gameplay requiring reactive and deliberative reasoning. The APM record holder is currently Park Sung-Joon with 818 (EVER Starleague [no date]). Outside of Starcraft APM is not as widely discussed but the importance of reaction time remains. Reaction time in AI is not something often discussed as many games opt to either hide actions to make a computer's instant response less obvious or cheat by applying delays to areas where players will definitely notice instant responses.

Another popular subject with AI is emotions. No one has succeeded in creating an AI with the emotions of a human yet but things that humans feel have started to be applied to games; most notably in the strategy game genre are Civilization (Firaxis Games 1991) and Total War (Creative Assembly 2000) where as previously mentioned the AI will feel trust or develop opinions of you based on your actions and take their emotions towards you into account before making decisions.

By far the biggest downfall for AI in strategy games is their ability to actually strategize outside of their predefined strategies. There is no strategy game AI out there yet that makes dynamic long term decisions, only AI that follow a long term decision tree that has been programmed into them such as in Hearts of Iron 4 (Paradox Interactive 2016) where the lack of freedom in the AIs long term decision making is masked by the game trying to recreate real historical situations (Rowley, J 2017). For example, Germany will develop forces and prepare for a war and then declare that war in the game in 1939, the same year that World War 2 started. The actions that Germany take in the game also somewhat replicate the events of World War 2 but if at this point in the game Germany is in a vastly different situation than the predefined storyline - if for example they no longer own Germany and have somehow acquired Sweden the AI simply doesn't know what to do as it has gone too far from its original predefined path causing the AI to return to the basic rule-base AI which can be easily exploited or simply break (Anon 2016c).

2.5.1 The Turing Test

The Turing Test is a test created by British computer scientist and mathematician Alan Turing in 1950 after posing the question "Can machines think?" and "If a computer could think, how could we tell?" (Turing, A 1950). Turing proposed a test where there would be one human interrogator and two or more players with at least one of them being a machine, the interrogator must figure out which player is machine by chatting over a chat system while the machine attempts to be as indistinguishable from a human as possible; if the interrogator is unable to tell which player is machine the machine passes the Turing Test.

"A computer would deserve to be called intelligent if it could deceive a human into believing that it was human." (Turing, A [no date])

Turing's experiment has lead to many interesting AIs being developed with the sole purpose to pass the test such as the first claim to success "ELIZA" (Weizenbaum, J. 1966) with a fairly small script managed to mislead interrogators by making them talk more and using the interrogator's own questions against them, other examples include "PARRY" (Colby, K. 1972) who imitated a paranoid schizophrenic who would attempt to steer the conversation back to its preprogrammed obsessions. Formal competitions of the test are held annually by The Loebner Prize (Loebner Prize 2015) with prizes handed out the most human-like AI each year; thus keeping the discussion active in the AI community. Whether the Loebner Prize and the Turing Test is an accurate representation of machine intelligence over trickery remains a debate (Hardawar, D 2015). The test has more recently been a talking point in mainstream media with the release of Ex Machina (Ex Machina 2015) which follows a programmer invited to conduct the Turing Test on a humanoid robot with a series of underlying puzzles and even a hidden test (Anon 2016a) for the audience (Anon 2016b). Even more recently the game “The Turing Test” (Bulkhead Interactive 2016) which is said to include puzzles only solvable by humans having us once again questioning, what does it mean to be human?

2.6 Fuzzy Logic

Fuzzy Logic is a form of AI decision making which deals with degrees of truth rather than a binary decision. Most often credited to Prof. Lotfi Zadeh of the University of California at Berkeley in his 1965 paper on Fuzzy Sets (Zadeh, L 1965) where he described fuzzy logic as a way to allow machines to solve problems similar to how a human might approach it. As pointed out in the paper Metamathematics of Fuzzy Logic (Pelletier, F 2000) the true origin of the logic system can be traced back to the 1920’s and 1930’s where Jan Łukasiewicz and Alfred Tarski first investigated the same logic at the time named infinite-valued logic; however, Prof. Zadeh should be credited for the logic system being re-popularised and making it a viable tool for AI systems.

Fuzzy Logic allows the AI to evaluate the amount of yes and the amount of no rather than the simple binary decision of yes or no.

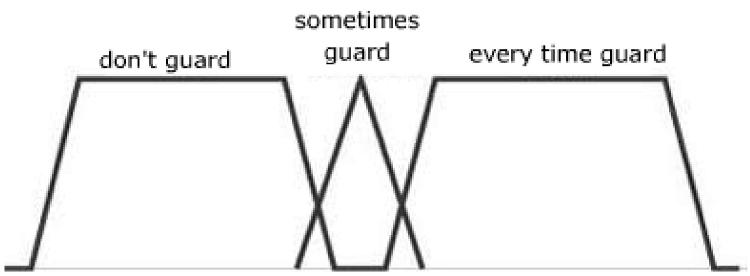

Figure 5 - *Fuzzy Set* (Ertürk 2009)

Figure 5 presents a common fuzzy set for a simple fuzzy logic problem. A fuzzy system allows us to have several inputs such as ammo, health, distance from enemy and output a graph that represents the degrees of truth to whether the AI should guard, sometimes guard or never guard. Standard Fuzzy Logic (SFL) Sets takes these fuzzified inputs and forms the graph presented in Figure 6. After forming the graph another calculation is performed to calculate the centre of gravity, or centroid; the final output often being the x-axis value of the centroid, giving the system the decision to be made Figure 6 shows the end result of a fuzzy calculation, which if applied to the decisions of Figure 5 results in the AI deciding “don’t guard”.

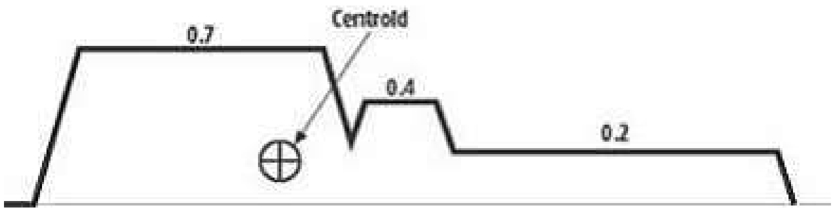

Figure 6 - *Fuzzy Output* (Bourg and Seemann 2004, p.204)

Although fuzzy logic without a doubt makes an AI less easy to predict, the AI will for the most part continue to follow the basic rules that would be laid out in a similar rule-based system when reduced down to its crisp binary output. The SFL system should rarely make the wrong decision while a human may not be as precise; so how do we add this possibility to make the wrong decision without deliberately making the AI dumb? A possible solution for this is to use the truth values for the fuzzy set as probability values. With the values displayed in Figure 6 and a simple calculation of dividing the truth value by the total amount of all truth values gives us 54% chance of “don’t guard”, 31% chance of “sometimes guard” and 15% chance of “every time guard”. This approach to the decision making serves as an alternative to the SFL system which can always be reduced down to its crisp output; allowing us to have an AI that makes logical mistakes.

2.7 Neural Networks

Although out with the project's scope, an important AI technique to cover is Artificial Neural Networks (ANN) as they are considered to be the closest we are to the architecture of how humans think. The understanding so far is that a neuron takes a great number of inputs and then uses this information in a complex algorithm, known as the “Hidden Layer” and then output a decision represented by pulses, this output is carried to other neurons which can repeat this process thousands of times until a decision is made. Figure 7 shows a simplified neural network. This model of how we believe humans think has assisted game programmers in creating AIs like in the Creatures game series (Millennium Interactive 1996) with critical advantages over other AI techniques. Although its applications are still to be fully explored the possibility of using neural networks to create an AI that can think like a human has been a central talking point for years.

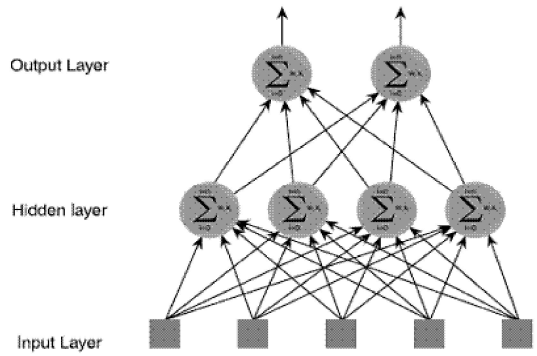

Figure 7 - *Structure of a Neural Network* (Buckland 2002, p.242)

2.8 Hybrid AI: Neuro-Fuzzy

Hybrid intelligent systems take a combination of AI techniques and deploys them in parallel; either working together or performing tasks in different areas of the decision making process. Fuller states that hybrid systems are being applied in many areas and proving to be a great success (Fullér 1995, p.207); allowing the best of each technique to contribute to the goal of a more advanced AI system. The hybrid intelligent system of interest here is the combination of neural networks and fuzzy logic, named Neuro-Fuzzy. Although neural networks allow us to have AI analysis patterns and perform complex decision making much like a human and fuzzy logic provides an effective way to engineer less easy to predict decision making, each have their disadvantages when it comes to games AI. Fuzzy logics weakness being it is essentially a more mathematically advanced way of performing a binary decision and more often than not, due to the technical limitation of performing such mathematical equations repeatedly it is inefficient to use fuzzy logic for long term decision making. Artificial Neural Networks (ANN) most obvious problem is that it is simply overkill for decision making where a simple rule-based solution would suffice. In addition to this ANNs consume massive amounts of resources and even for simple tasks become extremely hard to debug.

The possibility of having each technique be applied to specific areas of the AI where they are most effective may be the key to unlocking the true potential of AI systems. Although no known game has tried this approach, Umut Riza Ertürk theorized a system where Fuzzy Logic handles short term decision making and Neural Networks the long term decisions. Ertürk believes “the result for fuzzy logic seems to fit like a glove [for] the given problem.” (Ertürk 2009) and although no experiments were able to be carried out due to limitations in hardware Ertürk concluded “conceptual ANN distributed systems can be a new approach for solving the problems of ANNs in RTS games.” (Ertürk 2009) in relation to long term decision making. While unfortunately outwith the scope of the project the research conducted here may help reinforce the findings of Ertürk and hopefully inspire later generations to develop such a system.

3 Methodology

This section will provide an overview of the practical work completed for this project including details of the developed AI systems and real-time strategy (RTS) game environment in-which they were tested. All scripts (excluding FPSDisplay.cs and scripts located in the standard assets folder, where only TiltShift.cs has been used) and scenes included in the game were created solely by myself for this project.

3.1 The Game Environment - Little Planets

3.1.1 Inspiration & Overview

Figure 8 - *Little Planets Title Screen* (2017)

The developed RTS game environment, Little Planets, was developed after inspiration from the Google AI Challenge in the Autumn of 2010 named "Planet Wars" (Melis, G 2011), where on an open map representing the galaxy there are planets on a 2D plane. Each planet, if owned by a player, would spawn units to claim a planet the player must send units from their planet to either neutral (unowned) or enemy planets; if the units sent to claim a planet outnumber those defending it, the planet is claimed by the attackers. In the Google AI Challenge several AIs would play against each other by first starting at their home planet and taking over surrounding neutral planets and then must use unit management and other strategies to take other AI planets and eventually be the last remaining AI to win the game. This is a simple game idea based on Galcon (Hassey, P 2008); since the release of Galcon many games have been created with slight adaptions on how you play, most notably and the biggest inspiration for my project is Little Stars for Little Wars 2 (MKG 2012) shown in Figure 9, where they implemented multiple planet upgrade types and connected planets rather than allowing any planet to travel to any planet.

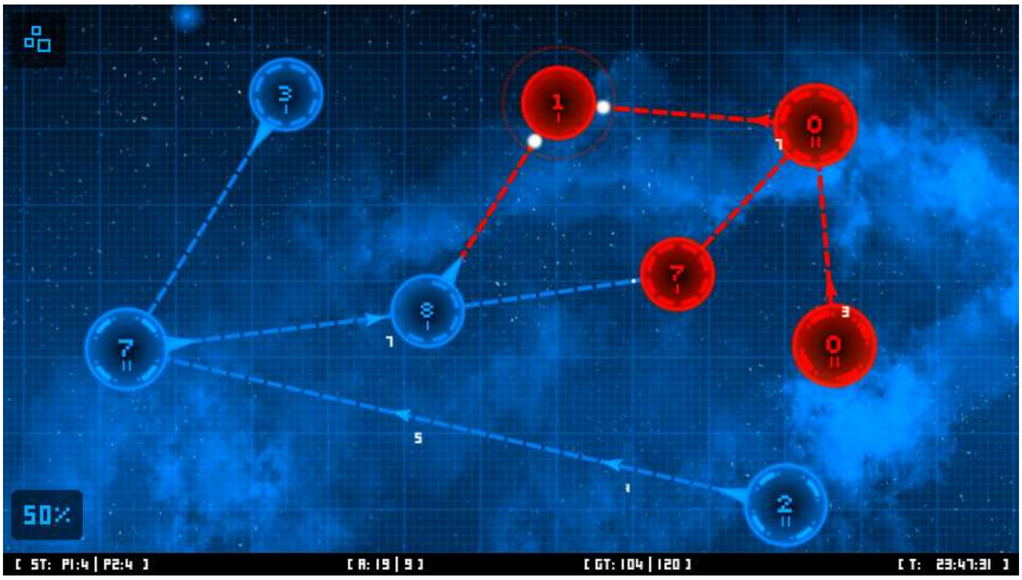

Figure 9 - ***Screenshot of Little Stars for Little Wars 2*** **(MKG 2012)**

This game design is both simple to play and understand but also challenging for players and AI alike as they will have many factors to consider before committing to any actions. Figure 8 and 10 show the title screen and in-game screenshot of Little Planets respectively.

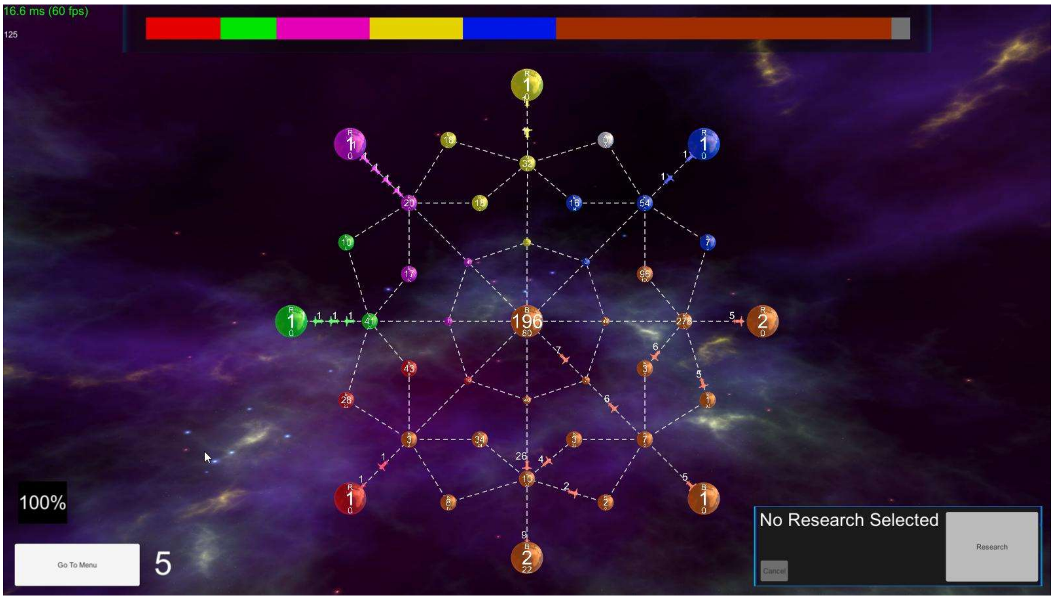

Figure 10 - *Little Planets Screenshot* (2017)

3.1.2 Additional Features

The game design lends itself well to modular game design, meaning no mechanic relies on another and there are plentiful mechanics that can be added to make the game more complex. This allows me to easily remove features which would be too out with the scope of the project and too complex to write an AI for, such as Fog of War and easily add features to help demonstrate the human-like behaviours of the AI developed. Little Planets' main features include:

Planetary upgrades; where each upgrade has pros and cons, adding an extra layer of decision making. Players can identify the upgrade on a planet by the letter shown above the planet's units as shown in Figure 11.

Figure 11 -*Planet Upgrades in Little Planets* (2017)

Multiple planet sizes; with the general idea that bigger is better allows players and AI to prioritize their movements across the map and helps enforce power struggles over the larger planets within each map.

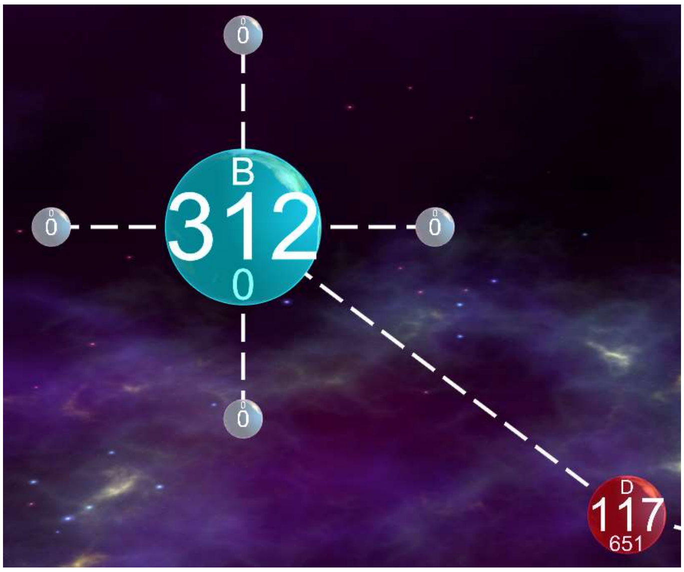

Figure 12 -*Planet Sizes in Little Planets* (2017)

Auto Deploy; a unique research within the game that only needs researched once. Auto Deploy allows players (human and AI) to set a automatic route from the planets they own, visually identified as an arrow pointing from the planet to its target such as in Figure 13, every couple seconds the planet will automatically send all units from the planet towards the target route.

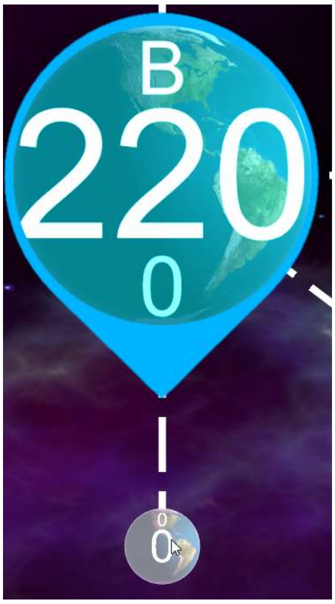

Figure 13 - *AutoDeploy in Little Planets* (2017)

Research; similar to how games such as Stellaris (Paradox Interactive 2016) handle research, you have multiple areas of research including Ship Construction, Shield Generation, Warp Drive and the unique one-time research AutoDeploy where, other than AutoDeploy, each time the research is completed increases the attribute associated with the research; for example, upgrading Warp Drive will increase the speed at which ships travel. All upgrades start at level 0 and in order to unlock the mechanic you must at least have the research at level 1; for example, if Shield Generation remains on level 0, shields will not generate.

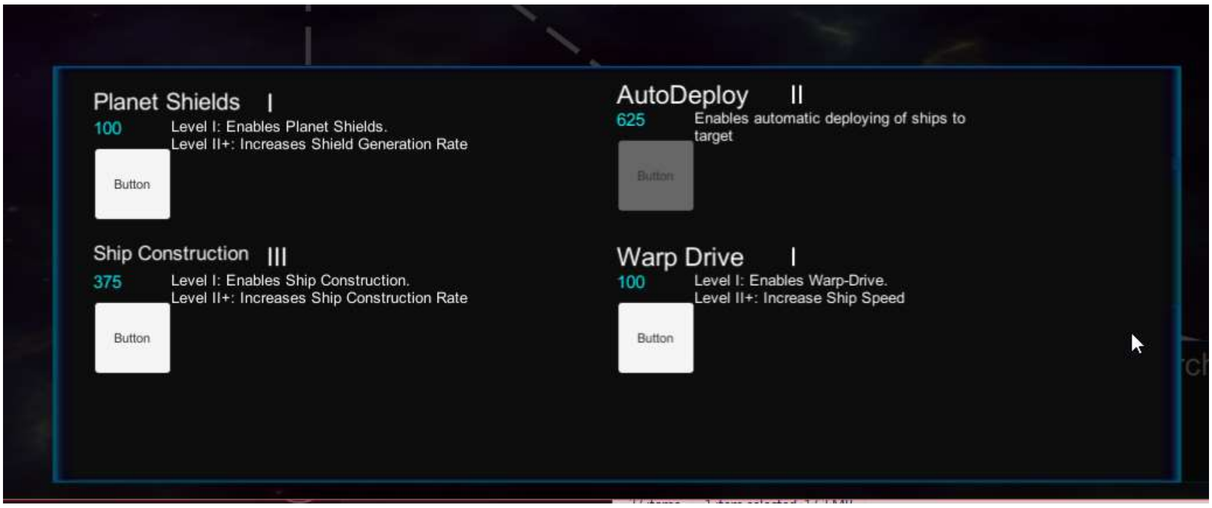

Figure 14 - *Research Interface in Little Planets* (2017)

Factions and Diplomacy; although all examples of the Planet Wars game concept have different colours to represent each player, other than their visual appearance the colour for the most part did not matter, if you were a different colour you were an enemy; it is a strong personal believe that this lack of exploration into the "faction" system is what is holding back the potential of this type of game so Little Planets offers a way to communicate with other factions either by setting them as your rival or offering them an alliance. The main way of learning information about each faction is via the "Balance of Power" element as shown in Figure 15 that shows the influence of each faction in relation to how many planets they own out of the total available. Hovering over a colour provides additional information, as also shown in Figure 15. Clicking on a faction's colour opens the diplomatic menu shown in Figure 16.

Figure 15 - *Balance of Power in Little Planets* (2017)

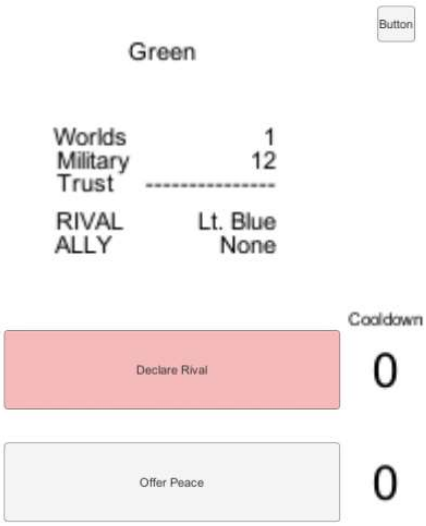

Figure 16 - *Diplomatic Menu in Little Planets* (2017)

Notifications; although not directly related to the AI, I spent a bit of time implementing my own type of notifications in two forms. Push Notifications appear in the middle of the screen and can have either an “OK” option or “Accept” / “Decline” for when AI offer the player an alliance as shown in Figure 17.

Figure 17 - ***Push Notification in Little Planets*** (2017)

The second notification type is Temp Notifications; these appear at the side of the screen as seen in Figure 18 and disappear after a couple of seconds. Temp Notifications, are used to show when an AI declares a rival or an alliance is formed; however, during the debugging phase of developing the AI Temp Notifications served great use in seeing when an AI performed an action such as spawning ships or upgrading a planet.

Figure 18 - ***Temp Notifications in Little Planets*** (2017)

3.2 Developed AI

The game design lends itself extremely well to a simple set of rules which can govern how the AI should generally act, this was a great benefactor in the creation of these AI systems as they all followed this similar rule-set with the key difference being how they came to their decisions. Throughout the following figures in this section areas which are accessed only by the AI that use the “Human-Like” qualities including: Reaction Time, Trust and Diplomacy; will be highlighted in yellow - These AI will also be referred to a “Simple” Systems. The Simple Fuzzy Logic and Simple Rule Based AI do not have these features and simply skip these steps.

A total of 5 AI seperate AI systems were developed:

1. S_RuleBase (Simple Rule Base);
2. A_RuleBase (Advanced Rule Base);
3. S_FuzzyLogic (Simple Fuzzy Logic);
4. A_FuzzyLogic (Advanced Fuzzy Logic);
5. SA_FuzzyLogic (Probability Fuzzy Logic).

Each technique will be discussed in more detail later in this section.

3.2.1 AI Controller Structure

Before developing the AI the basic structure's layout and rule-set must be set. Due to several factions having the possibility of being one of several AI a state machine was created with the states each AI type. This script would cycle through each faction checking if the faction was still active in the game (owns at least one planet) and to check if it was not the human player. Having the code structured like this allows adjustments to each AI rather than having general functions that one script would call. The basic structure behind AI_Controller.cs is detailed in Figure 19.

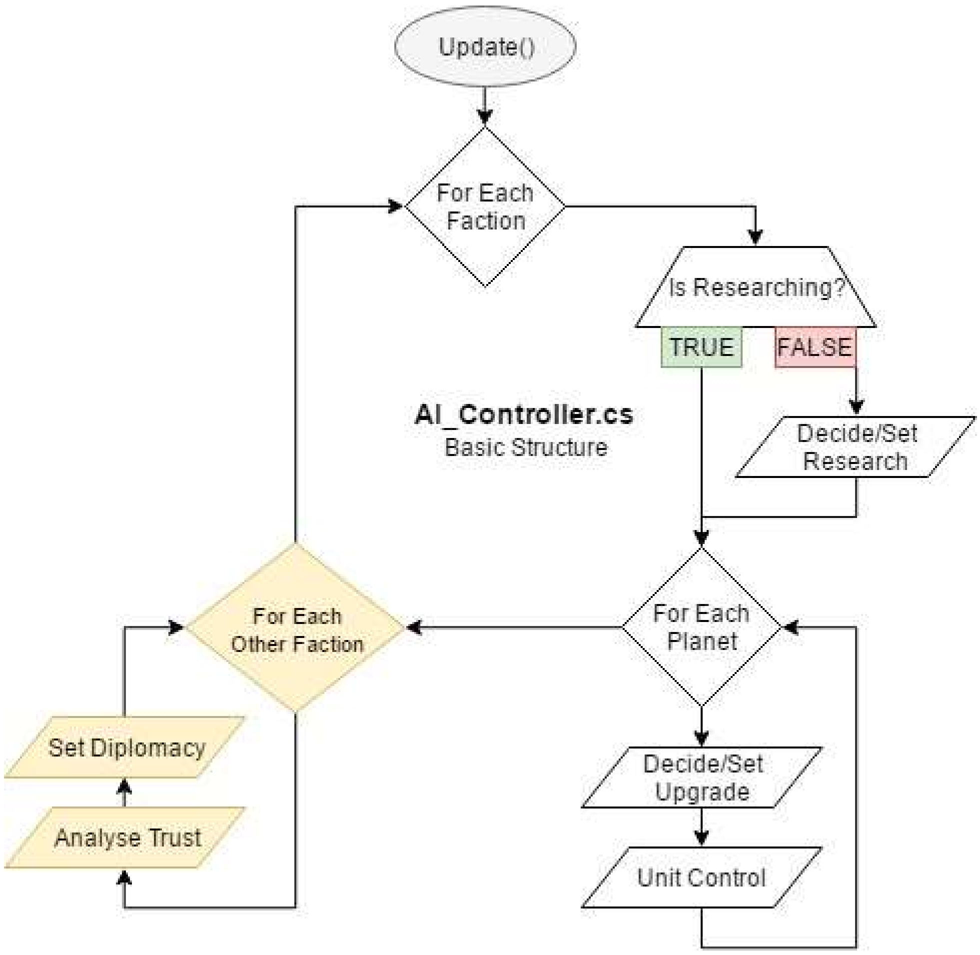

Figure 19 - *AI_Controller.cs Structure*

Every frame the AI would cycle and perform the actions it needed to; however, after developing this first AI it became clear that having the AI think and decide every frame was extremely unfair and made it almost impossible for the human players to win so a standard delay that all AI have was implemented which will be discussed later. The AI performs a vast number of checks to ensure simple things such as if a research is already max level or simple unit number checks (to ensure it has enough to perform the action); these will not be detailed in structure figures.

3.2.2 Rule Base Systems

The rule-based AI takes the simple rules already defined in the games this project is based on and applies them to this game.

3.2.2.1 Research Decision

The first step performed by any AI is whether or not it should being a research, if it should it then decides which research based on a number of rules which are detailed in the structure of the decision making in Figure 20.

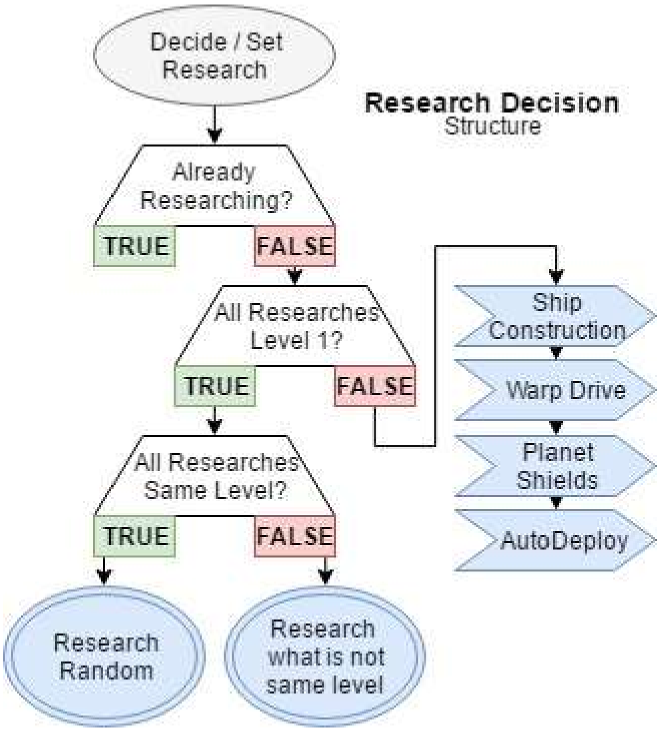

Figure 20 - *Research Decision Structure*

3.2.2.2 Upgrading Planets

The next step for the AI is to decide if a planet should change its upgrade type. A type can only be changed if the planet has enough units (defaulted to 10). The following decision structure displayed in Figure 21 is applied:

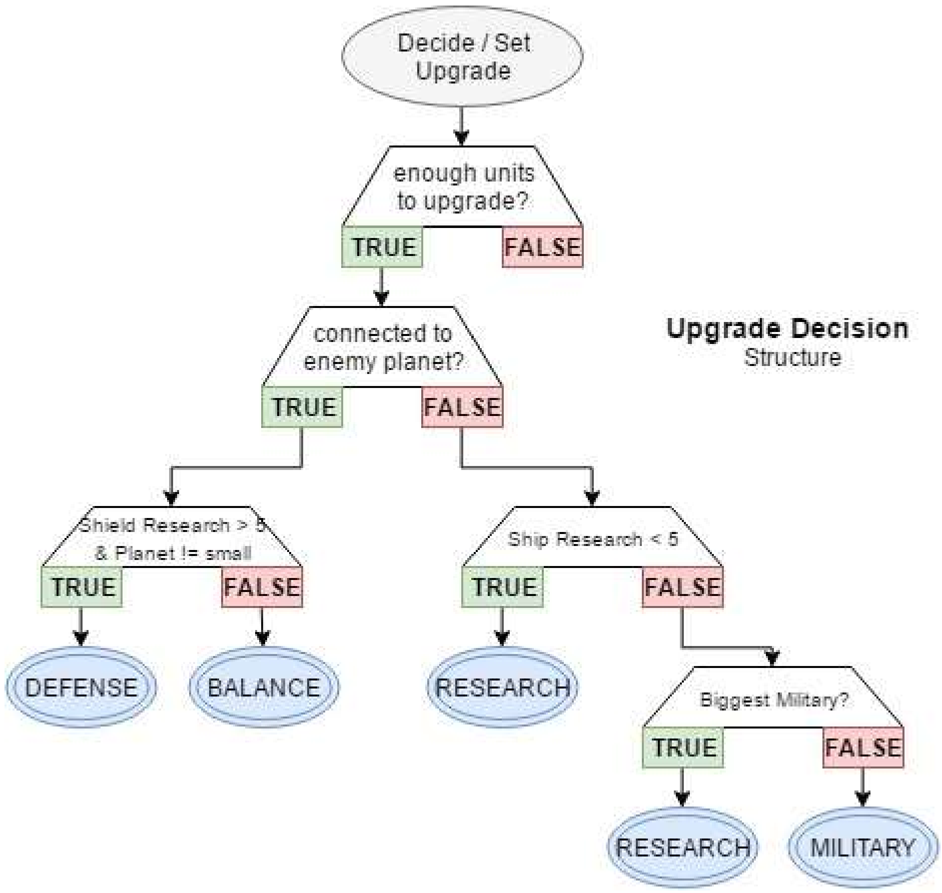

Figure 21 - ***Upgrade Decision Structure***

3.2.2.3 Unit Control

All AIs then enter the Unit Control phase, this phase dictates what the AI will do with the units in a given planet. Figure 22 details the structure of the decision making. AIs can send any amount of available units from the planet while making these decisions to the closest 10 percent - human players also have this ability. Most commonly the AI will send just enough units to perform the action; however, sometimes often based on skill level the AI can send all of its forces or a slightly higher percentage to ensure the attack is successful and secure.

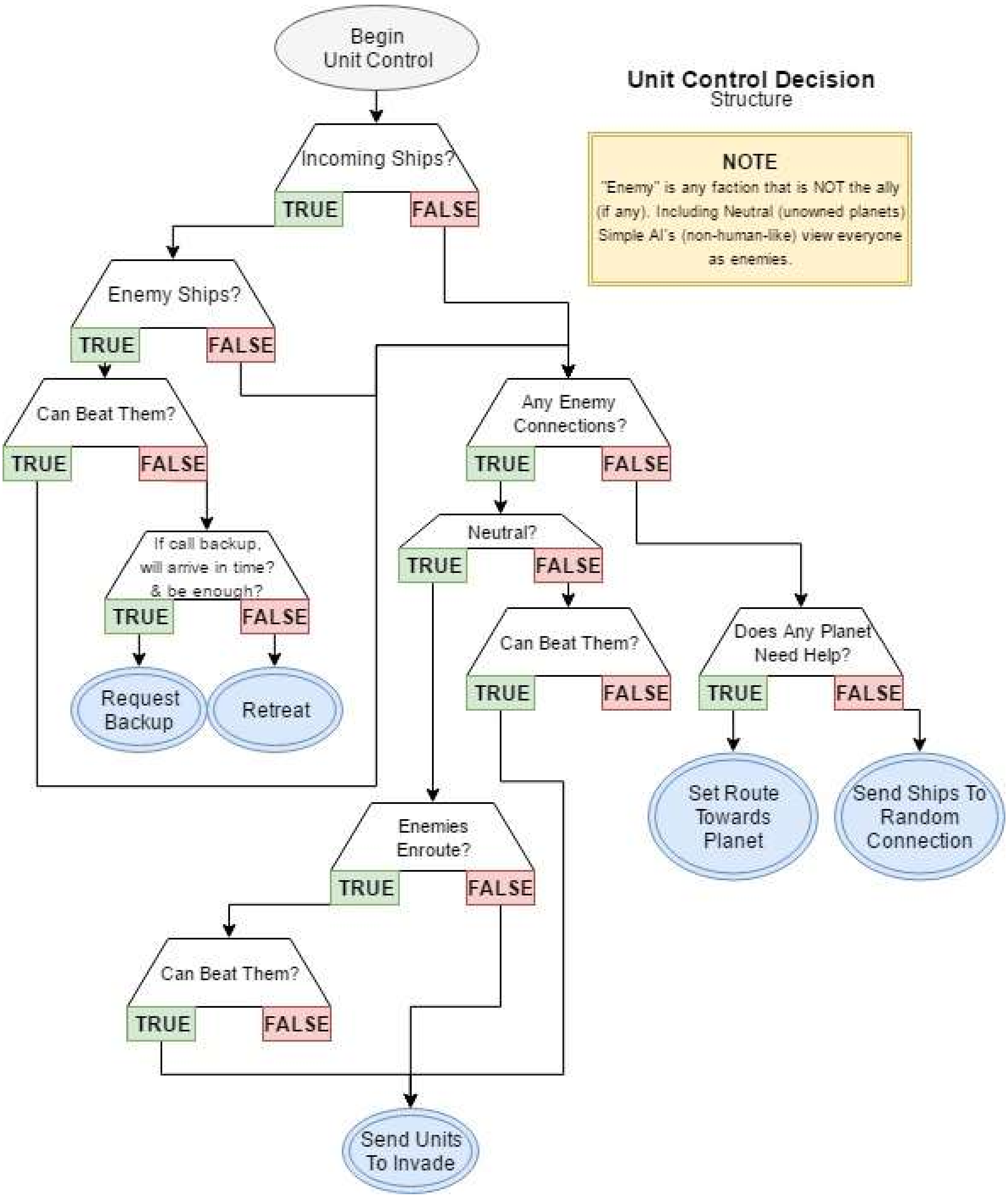

Figure 22 - *Unit Control Decision Structure*

3.2.2.4 Trust and Diplomacy

Note that this section is skipped by the “simple” AI techniques that do not feature the human-like qualities including the Simple Rule-Base and Simple Fuzzy Logic systems.

This was by far the hardest quality to implement due to the complexity of the balancing of the positive and negative effecting variables that had maximum and minimum values. For example, the faction's military power would affect the trust value on a scale between negative and positive 10. This became a strain on time again due to the amount of detail required for balancing such a system. Instead the decision was made to implement a messier but slightly easier to balance approach of having the trust value towards each faction constantly ticking up or down each run of the loop by a manipulation value which is effected by a dynamic amount relating to the thing affecting the trust. An example of this is one of the first checks which is if the military power of a faction is greater than the current AI; if it is, the AI takes the percentage difference between their military power and the other faction and decreases the trust towards that faction by this amount. A similar process is repeated for many aspects of the game including number of owned worlds and time since attacked. Figure 23 shows the in-game trust window used to help debug and balance the trust values.

Figure 23 - *Trust / Diplo Stats in Little Planets* (2017)

Publically declared rivalries within the game also play a role in the trust towards a faction, if a faction shares the same rival as you their trust towards you increases greatly; on the flipside, if your rival is another factions ally they will recognize this and your trust with them will rapidly decline. Due to not being able to fully control these values limitations were put in place to ensure the trust between factions would never fall below -500 or go above 500, this helped keep the trust value within a manageable range for balancing.

The next step is to create values to store the desire the AI has towards each other faction, similar to how trust is manipulated desire values make sure of dynamic variables including military power, research levels, owned worlds and trust. Using these dynamic variables ensures the maximum and minimum values are known so the system can be properly balanced. As previously mentioned this technique is superior but very time consuming in terms of balancing; however, the previous solution cannot be applied for the “desire” values. An example of a desire value uses the trust value (which can only range between -500 and 500) and divides it by 100 making the desire value range between -5 and 5. There are some hard-set values related to whether the faction is a rival or already has an ally to ensure the AI acts accordingly since an AI should not want to be allies with a rival unless it was extremely beneficial based on the other desire values.

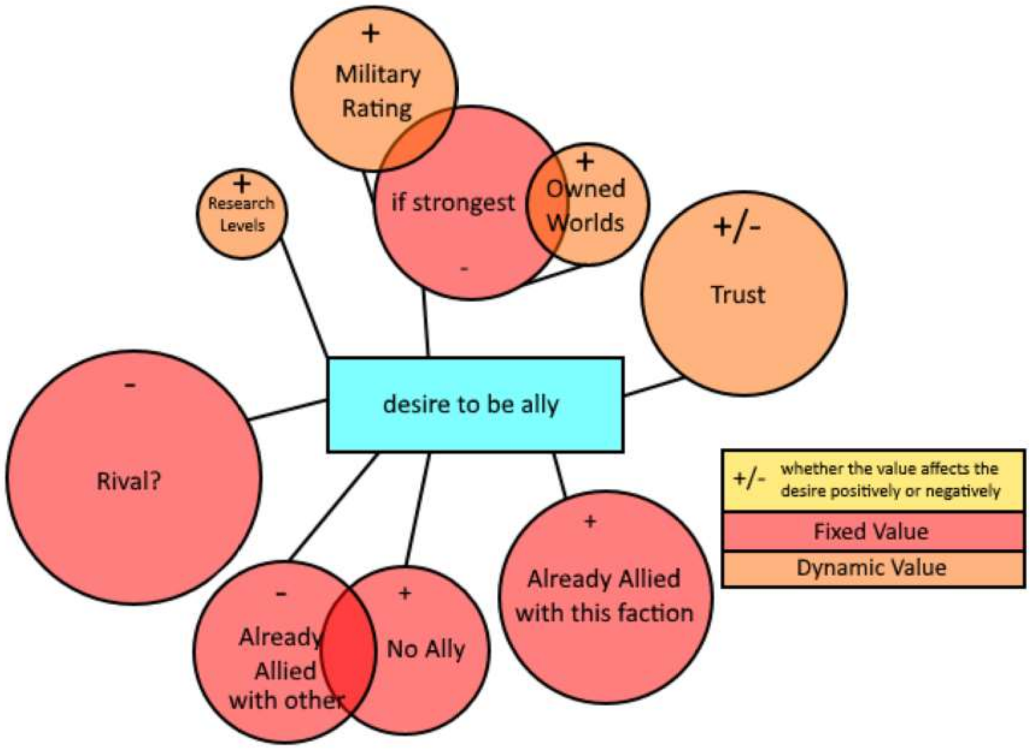

Figure 24 - *Visual Representation of Values Affecting Desire*

3.2.3 Fuzzy Logic

Standard Fuzzy Logic (SFL) as explained in Section 2.6 uses a fuzzy input to generate the fuzzy set and then calculates the centre of gravity of the resulting graph from the Fuzzy Set. For a game designed like Little Planets, fuzzy input was believed to not be the correct form of implementation due to the nature of the rules. Instead, with inspiration from neutral networks and investigation of other systems online, this game lent itself better to custom made weight values to dictate the influence of each decision within the Fuzzy Set. These influence values are calculated using information that the AI has available that relate to the rules defined in the rule-based systems. Influence values are always between 0.0 and 1.0, representing 0 to 100%; these influences are then multiplied by the weight they carry in the decision making and added together to produce another total influence value for that decision which again lies between 0 and 1. This value finally passes through a self-made script: FuzzyMath.cs.

3.2.3.1 Research

Similar to the other AI, each fuzzy logic system begins by deciding which research to choose from. While developing and calculating the degrees of truth/influence towards each research it became increasingly clear that using a fuzzy logic set as pictured in Figure 5 would be inappropriate as there would be 3 decisions (excluding AutoDeploy). If for example the graph was laid out with the order of the areas being Ships, Shields, Warp and the influence of Ships and Warp were 0.8 no matter what value shields were, the decision would be to upgrade shields which makes no sense. So rather than use fuzzy sets for the calculations the code simply takes the influence towards each research (named: Wshield, Wunits, Wauto, Wwarp) and selects the one with the greatest influence. The equations 2, 3, 4 and 5 detail the calculations for each influence/weight, including the faction weight shown as Equation 1 which represents what percentage of the alive factions the AI represents.

$$factionweight = \frac{1.0}{aliveFactions} \tag{1}$$

$$Wshield = \frac{ownedNonSmallPlanets}{totalNonSmallPlanets \cdot factionweight} \cdot \frac{ownedNonSmallPlanetsWithShieldsActive}{ownedNonSmallPlanets} \tag{2}$$

$$Wunits = \frac{militaryRank}{aliveFactions} \tag{3}$$

$$Wauto = \frac{ownedConnections}{totalPlanetConnections} \tag{4}$$

$$Wwarp = \frac{totalConnectionsFromOwnedPlanets}{totalConnections \cdot factionweight} \tag{5}$$

3.2.3.2 FuzzyMath.cs

Thankfully the problem encountered during the Research weight calculations does not affect planet upgrades or unit movement. Therefore the self-made implementation of fuzzy logic was utilized.

This script creates a virtual fuzzy graph using a set of predefined values that can be applied to both Planet Upgrades and Unit Movement which will use this Fuzzy Set. Figure 25 shows the Fuzzy Set used.

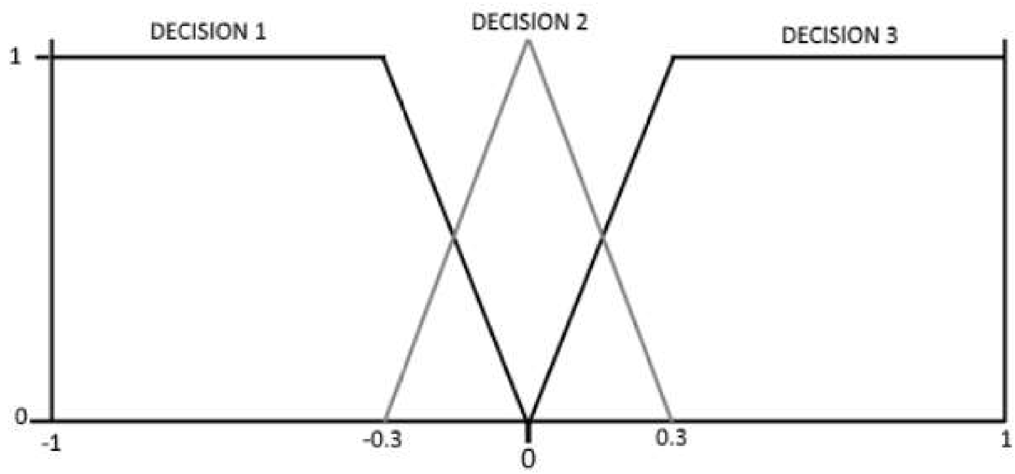

Figure 25 - *Fuzzy Set Created in FuzzyMath.cs*

As well as creating the set the script finds the centroid point using Equation 6 with example Figure 26.

$$COG = \frac{\sum_{i=1}^{6} x_i \cdot y(x_i)}{\sum_{i=1}^{6} y(x_i)} \tag{6}$$

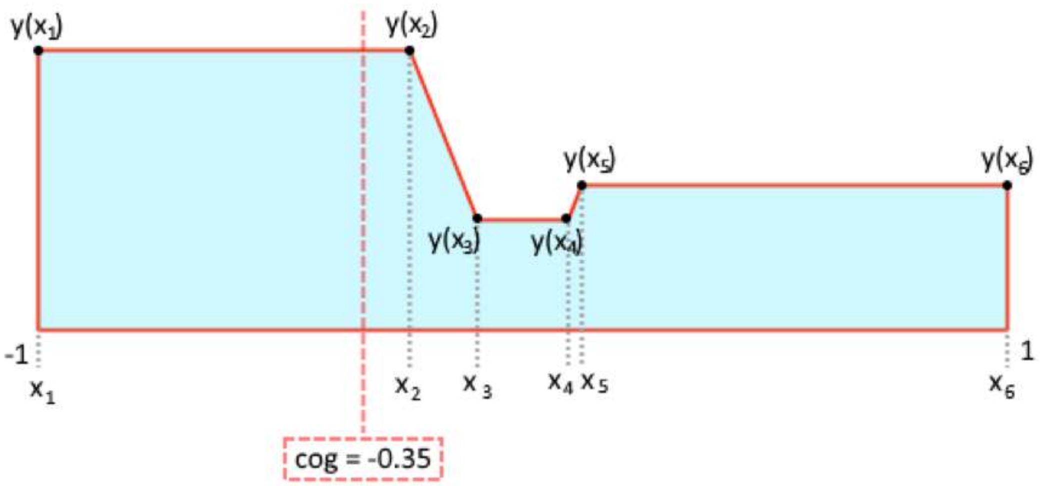

Figure 26 - *Fuzzy Set Output with Calculated Centre of Gravity*

3.2.3.3 Upgrading Planets

Upgrading features the three decisions of: Defensive, Balanced and Aggressive (Military) upgrades respectfully. For each decision are several contributing values, each with their own weights.

Defense

- Close Enemy - 50%
  - Using the distance an enemy planet is from the current planet. ie: closest enemy = 2 planets, means closeEnemy = 0.8 (40%).
- Shield Research - 50%
  - Takes the research level invested in shield generation and divides it by the the max level, meaning as the research levels up the chances of utilizing the upgrade that uses that research increases.

Balance

- Shield Percent - 75%
  - Current Shields divided by the maximum possible shield value, meaning the closer the shield gets to the maximum possible the higher chance there will be of the planet switching to a balanced version since the bonus of maximum shield generation is no longer needed
- Military Rank - 25%
  - Amount of units and planets compared to other factions, meaning the lower the military rank, the higher chance to upgrade to increase unit generation.

Military

- Military Rank - 60%
  - The lower the military rank, the higher the desire to upgrade the planet to produce more units
- Unit Research - 40%
  - The higher the Ship Construction research, the higher chance to utilize that research.

Research

The research upgrade is not included in the fuzzy calculation but instead its influence is calculated, if influence is stronger than all of the other upgrades research is chosen, if not it performs the fuzzy calculation.

- ❖ Close Enemy - 30%
  - ➢ If the planet is a safe distance from the enemy, higher chance to change to research.
- ❖ Military Power - 70%
  - ➢ The higher the military power, the higher chance to devote resources to researching since not as much pressure to produce units.

Each research has a series of rules in addition to the calculation of influence values that can either hard set the value to 100% or 0%. These are simple rules that make for absolute certain the desire to be this way; such as the research levels for all researches being max, therefore no longer a need for planets using the research upgrade; or if the planet is small there would never be any reason to upgrade to defense as small planets cannot have shields.

Once calculated the three decision value are passed into the fuzzy algorithm function FuzzyMath.Defuzz(x, y, z) to create and defuzzify the fuzzy set giving the output of either decision 1, 2 or 3.

3.2.3.4 Unit Movement

Defend:

- ❖ Incoming Enemies - 100%

OR, if no incoming enemies:

- ❖ Number of Enemy Connections - 50%

Pass:

- ❖ Number of Friendly Connections - 100%
  - ➢ The "Simple" AI, only take into considering connections owned by themselves to be friendly; whereas all other AI consider their set ally to be friendly.

Attack:

Attack Influence is calculated per connected planet and selects the connection planet with the highest attack influence to be its target and to carry forward to the fuzzy calculation.

- ❖ Percentage that Enemies Outnumber - 75%
- ❖ Neighbouring Enemies - 25%
- ❖ Reserves Limiter - Special Variable
    - ➢ This variable figures out the number of units that would need to remain on the planet in order to not lose it, if there is not enough units to successfully defend the Attack Influence is lowered by the % of how many units the planet is missing.

3.2.4 Probability Fuzzy Logic System

A proposed alternative to Standard Fuzzy Logic (SFL) is the use of the fuzzy logic sets and the influence for each decision has being used as probability. This proposed Probability Fuzzy Logic essentially allows implementation of an AI that can make logical mistakes. Suppose the example shown in Figure 27.

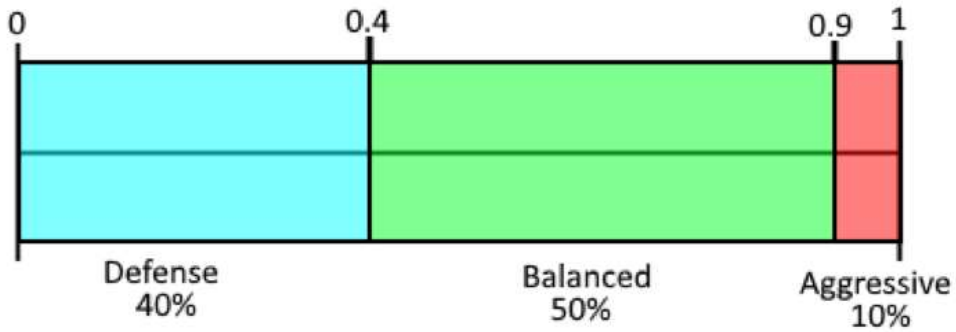

Figure 27 - *Probably Input*

The AI would select a random floating point value between 0 and 1, if that value were to land between 0 and 0.4 the defense upgrade will be chosen, if between 0.4 and 0.9 the balance and if 0.9 and 1.0 the military upgrade Thus giving each decision a logical probability of happening of 40%, 50% and 10% respectively. In practice this helps mimic the AI making a mistake. This successfully helps steer clear of the problem inherent in a SFL system although despite having a degree of unpredictability it will most likely still make the most logical choice. A problem encountered with the implementation of this way of decision making was that the AI would be too indecisive, as this update would be called perhaps every couple seconds the planets ended up changing focus every few seconds leading to the AI wasting units and it being painfully obvious that something was wrong with the AI. To combat this issue an additional weight value for each decision giving it +25% if it was already set; unfortunately although it helped the AI continued making too many changes to planet upgrades. To be clear, this method would work for the planetary upgrades given proper time to balance and test which was not possible for this project. Although the code remains, it is commented out and replaced with the regular custom fuzzy logic. With that said, the Probability Fuzzy Logic is still applied to the Research and Unit Movement decision making.

3.2.5 Human-Like Qualities

In addition to adding a degree of unpredictability to the fuzzy logic further implementation of human-like qualities was needed to not only make the same AI believable as a human but add increased variation of how the AIs approached problems and how quickly or skilled they were in doing so.

3.2.5.1 Skill Rating & Reaction Times

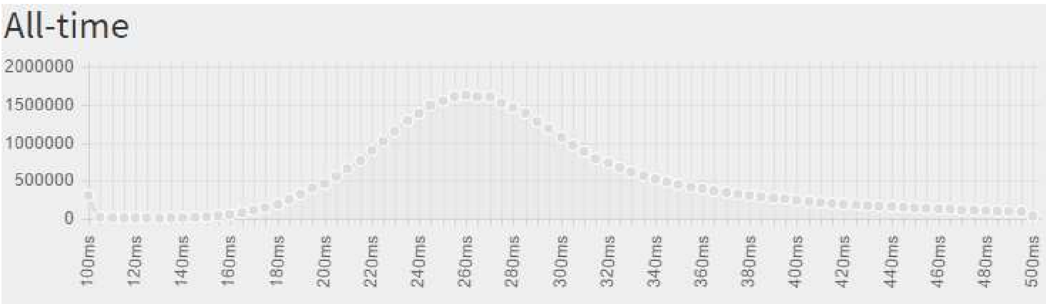

Figure 28 - *All-Time Human Reaction Times* (HumanBenchmark.com 2017)

Figure 28 displays the all-time data collected by HumanBenchmark.com (Human Benchmark 2017) where users can go to test their reaction times. Using this data alongside the general information given by the skill ratings of StarCraft II (Blizzard Entertainment 2010) players gives the approximate differences in reaction times, APM/APS (Actions Per Minute/Second) between the ranks ranging from complete beginner to the best in the world. Table 1 shows this collection of data and Table 2 the averages. All compiled research on APM and reaction time can be found in Appendix C.

Table 1 - General *APM, APS and Reaction Time Per Skill Level*

Skill Level	APM (min)	APM (max)	APS (min)	APS (max)	Reaction Time (max)	Reaction Time (min)
Noob	20	50	0.33	0.83	0.5100	0.475
Beginner	50	80	0.83	1.33	0.475	0.46
Gamer	80	180	1.33	3.00	0.46	0.41
Semi-Pro	180	280	3.00	4.67	0.41	0.36
Pro	280	400	4.67	6.67	0.36	0.3
Elite	500	800	8.33	13.33	0.25	0.1

Table 2 - *Average APM, APS and Reaction Time Per Skill Level*

AVG	APM	APS	RT
Noob	35	0.58	0.4925
Beginner	65	1.08	0.4675
Gamer	130	2.17	0.435
Semi-Pro	230	3.83	0.385
Pro	340	5.67	0.33
Elite	650	10.83	0.175

This data, although now more general, is still applicable only to StarCraft II; to convert this data to more suitable numbers for Little Planets several testers performed a number of tests with the Unity Engine (Unity Technologies 2005) timing reaction times through some simple scripts while performing some tasks within the game; all testers would consider themselves around the “Gamer” level so that was used as a standard for the testing. The first step was to calculate the differences in APM/APS and reaction time of the data in Table 2 which gives us the values shown in Table 3.

Table 3 - *Average APM, APS and Reaction Time Per Skill Level*

Noob	-12%
Beginner	-7%
Gamer	0%
Semi-Pro	12%
Pro	27%
Elite	85%

The second step was to then apply these differences between each skill from our median reaction time across tasks including Researching, Upgrading and Unit Movement as shown in Table 4.

Table 4 - *Compiled Minimum and Maximum Reaction Data*

Standard = Gamer						
Skill Level	RESEARCH (min)	RESEARCH (max)	UPGRADING (min)	UPGRADING (max)	SENDING UNITS(min)	SENDING UNITS (max)
Noob	0.475	5.619946092	0.475	2.247978437	0.475	3.371967655
Beginner	0.46	5.360110803	0.46	2.144044321	0.46	3.216066482
Gamer	0.41	5	0.41	2	0.41	3
Semi-Pro	0.36	4.390243902	0.36	1.756097561	0.36	2.634146341
Pro	0.3	3.62745098	0.3	1.450980392	0.3	2.176470588
Elite	0.1	0.737704918	0.1	0.2950819672	0.1	0.4426229508

This data was then implemented into reaction matrix via code storing the maximum and minimum time to perform a task. Coupled with the previously mentioned “Standard” reaction time these values were integrated into the AI_Controller only allowing the action to be performed if the timer for that reaction was below or equal to 0. If an action were successful it would reset the timer to a random floating point value between the minimum and maximum possible reaction times for that AI’s assigned skill level. This successfully drove away from the problem that the AI could upgrade every single planet and send units from every single planet every time it went through the loop in a split second of each other, now with this system the AI must prioritize which actions to be performed and if it cannot keep up, it will fall behind thus simulating the difference between professional and amatuer level play in real-time strategy games. This mechanic proved to be essential to not only the increased human-like qualities of the AI but also the balancing of the game, essentially giving the AI a delay to its actions; therefore, the standard reaction time has been applied to all AI controlled factions for the entry into the AI Controller state machine; for the Simple Rule-Base and Simple Fuzzy Logic AI this meant implementing a “FIXED” skill level to store a matrix of fixed reaction times which would not vary between minimums and maximums. It is worth noting that the implementation of this system dramatically improved performance as the AI was not always thinking.

3.2.5.2 Trust

As described in more detail in Section 3.2.2.4, each AI is able to develop trust based on a number of affecting factors. This system remains the same throughout all explored AI systems as it would not benefit from change per technique due to its nature. Trust is one of the mechanics that does not necessarily need to be represented in the game, thus trust values of AI are hidden from the player; the players only indicator of the trust an AI feels towards them is whether or not the AI takes the trust values into account when making decisions. One such example would be if the AI has the choice to invade two planets, each owned by a different player; the AI may decide to attack one planet over the other based on the trust it has towards either faction.

3.2.5.3 Strategy and Prioritisation

One of the challenges while developing the AI was finding some way for the AI to decipher which was the direction to the “objective”. Various pathfinding techniques were tried and tested but in the end a beacon system proved to be the most successful to implement into this sort of game. Essentially, a planet next to an enemy planet would set a number, 1000, to that planet; then each planet connected to that planet which has activated its beacon sets its beacon value to one value less than that planet, this repeats down all friendly planets resulting in a way for planets to identify which direction to send units by sending them to a connection with a greater number than its own. Figure 28 shows the game visualising beacons by increasing the glow around the planet as well as the help value above the planet - both of which disabled for human testing.

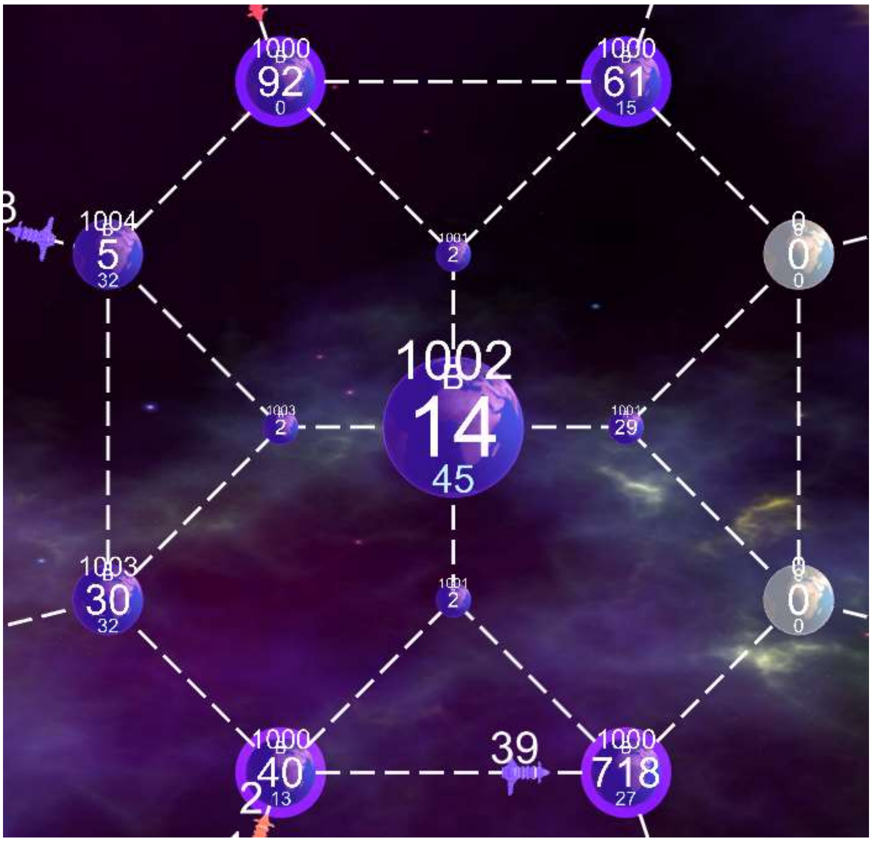

Figure 29 - ***Visual Indicators for Pathfinding/Help Values in Little Planets*** **(2017)**

This system proved itself extremely efficient but when it came to deciding which direction to go based on the position of your rival, the AI would gather forces on the opposite side of the frontline just because non-allied factions were next to it, despite there being little to know conflict in that area. To solve this planets that are beside rival factions simply set their help value to 1005 which successfully allowed the AI to move units between even the frontline planets in favor of reinforcing the frontline towards hostile, rival, planets.

3.3 The Turing Test

Unfortunately, multiplayer implementation became a bigger task than anticipated and in order to stand a chance of having it integrated into the game in order to perform a Turing Test would require more time than could be put in. So, instead of the traditional Turing Test, of having the test subject play against one (or several) human and one (or several) AI, the decision was made to opt for a form of Turing Test briefly described in this project's feasibility evaluation document where the tester is told they will be taking part in the Turing Test; the game then simulates searching for a multiplayer match with one other player. The implementation for this was fairly simple, with a short script TuringTest.cs and some visual indicators as shown in Figure 29 and Figure 30 the game successfully tricked all test subjects into believing they had joined a multiplayer session.

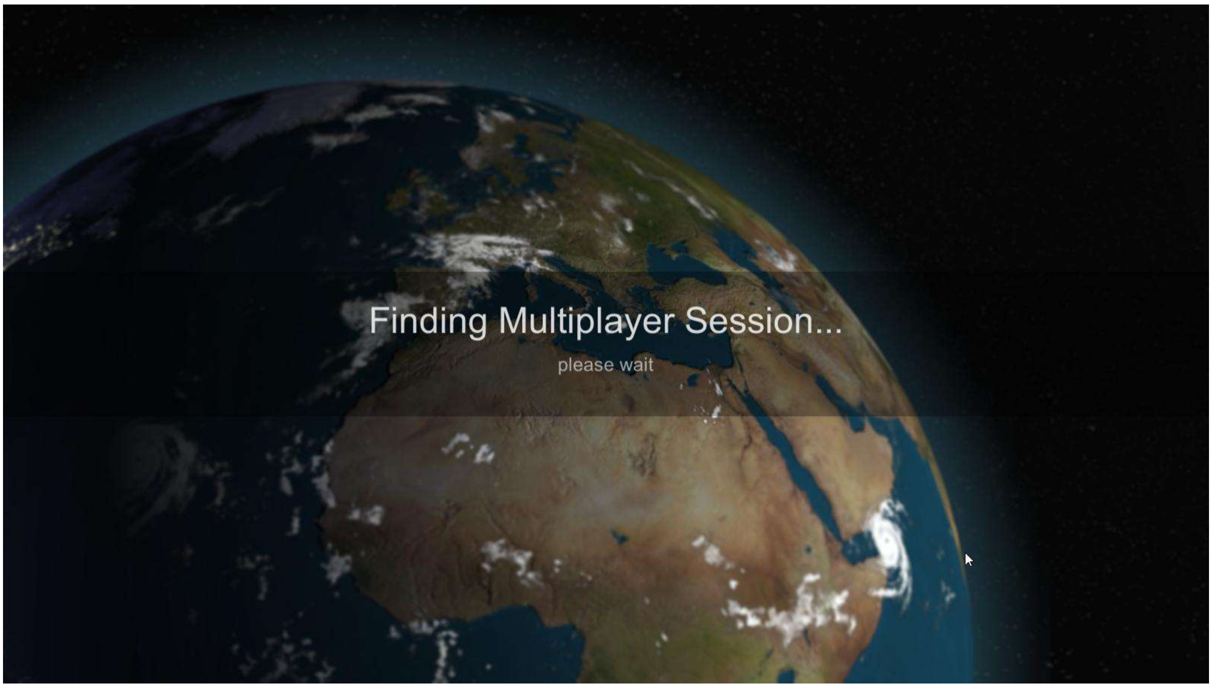

Figure 30 - *Searching for Multiplayer Session in Little Planets* (2017)

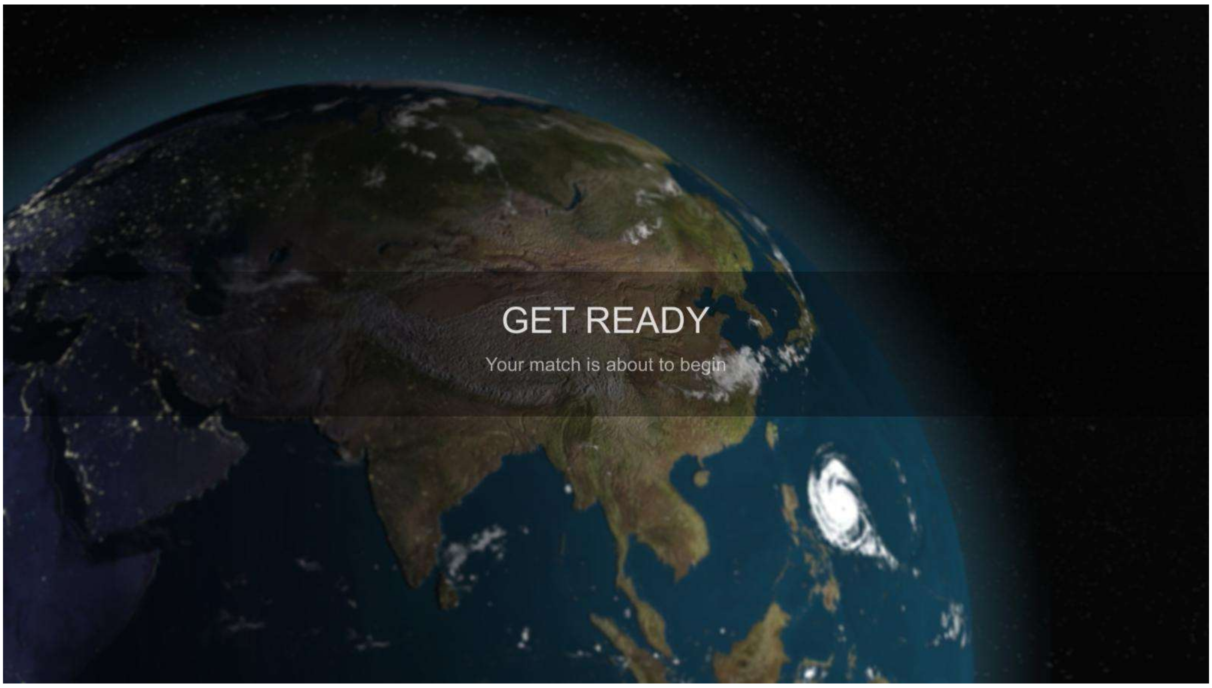

Figure 31 - *Connecting To Fake Multiplayer Match in Little Planets* (2017)

4 Results

In order to gather feedback and evaluate the user experience per AI type, a questionnaire and series of special game tests were prepared for 10 testers. The questionnaire began by gathering general information from the testers gaming habits to their opinion on AI and whether they preferred singleplayer or multiplayer. This data can be used in correlation with the testing results to gauge accuracy and thought that went into the testers' feedback scores; for example, a user who rated themselves high on a questionnaire regarding their interest and engagement in the conversation of AI may give more accurate results as they can detect subtle differences.

All questions that require an answer to a scale use the Likert Scale of 1 to 7. Testing was done remotely and unsupervised; although it is worth noting that only trusted and friends, family and colleagues responded, so there is no doubt in the results. Instructions were clear through testing and testers were reminded of important details throughout to ensure tests were being performed correctly.

In order to ensure all testers had some familiarity with the game time, all testers were asked to complete a tutorial within the game that goes through basic mechanics and then puts the tester in a practise match to allow them to get acquainted with the game. Full Questionnaire is available as Appendix A. Results as Appendix B.

4.1 General Survey

4.1.1 Previous Experience

Figures 32 to 36 displaces results of tester's previous experience with video-games and AI.

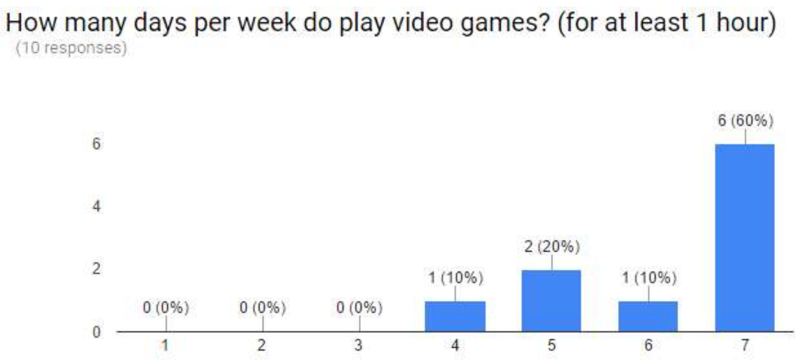

Figure 32 - *Results for Days Played Per Week*

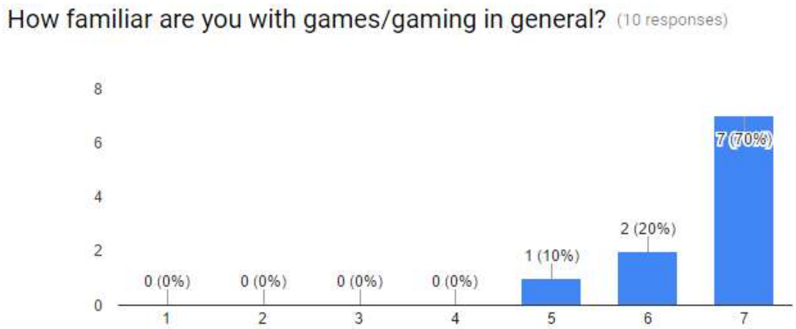

Figure 33 - *Results for Game Familiarity*

Figure 34 - *Results for Real-Time Strategy Familiarity*

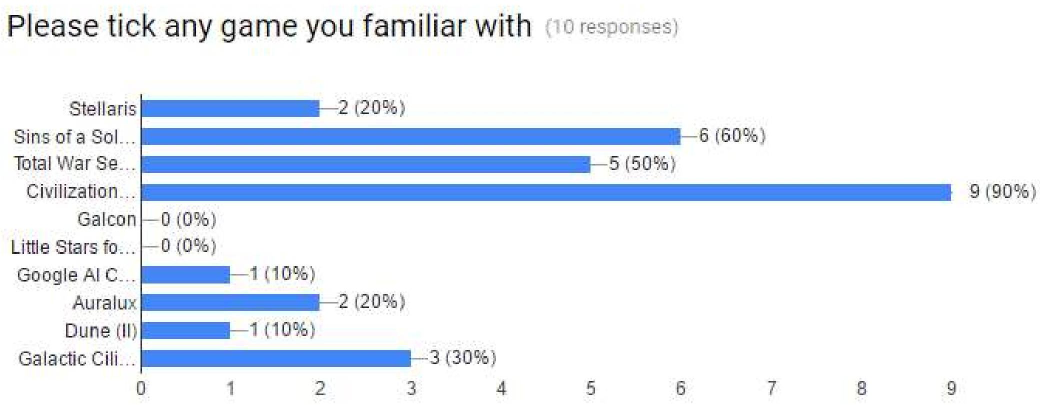

Figure 35 - ***Results for Awareness of Referenced Games***

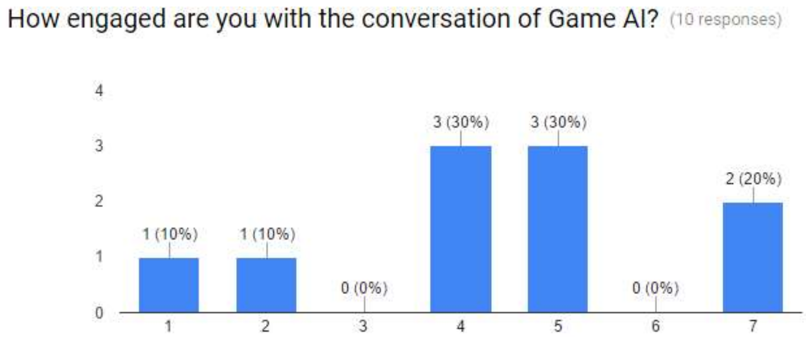

Figure 36 - ***Results for Engagement in AI Conversation***

These results are well spread with the exception of Figure 32 and Figure 33; this indicates that results will be focused from an avid gamers point of view and no results from casual or non-gamers were collected. This issue is minor as strategy games are definitely not a casual or non-gamers genre of interest, it simply means there will be no data available from the point of view of someone less experienced with games.

4.1.2 Opinions

Figures 37 to 41 display tester opinions.

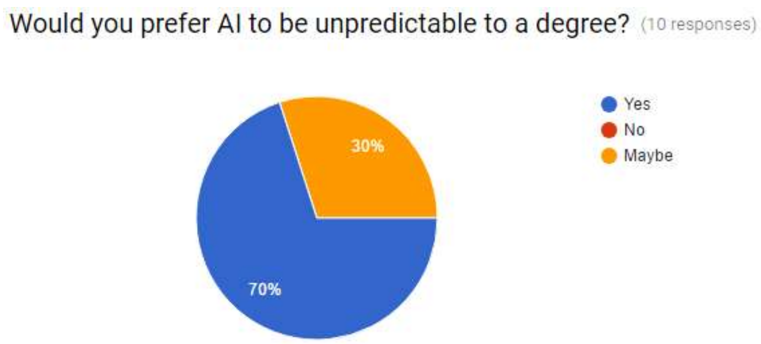

Figure 37 - *Results for Preference on AI Unpredictability*

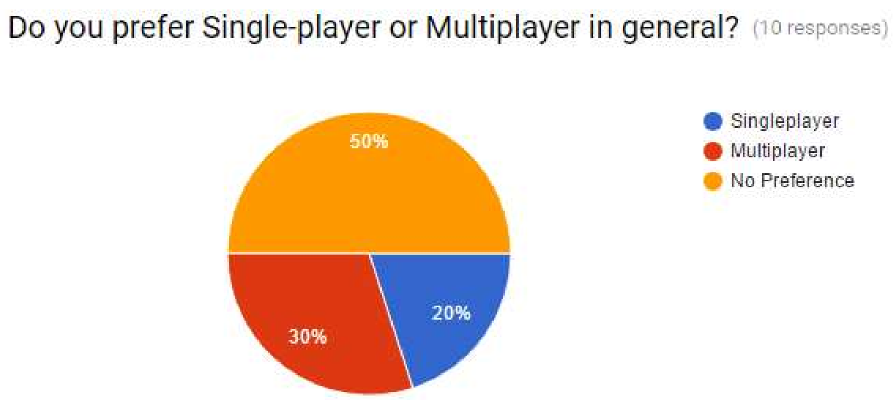

Figure 38 - *Results for Singleplayer vs Multiplayer Preference*

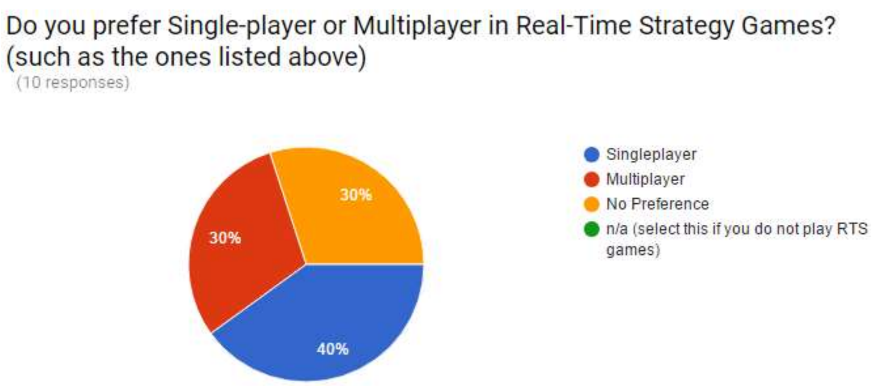

Figure 39 - *Results for RTS Singleplayer vs Multiplayer Preference*

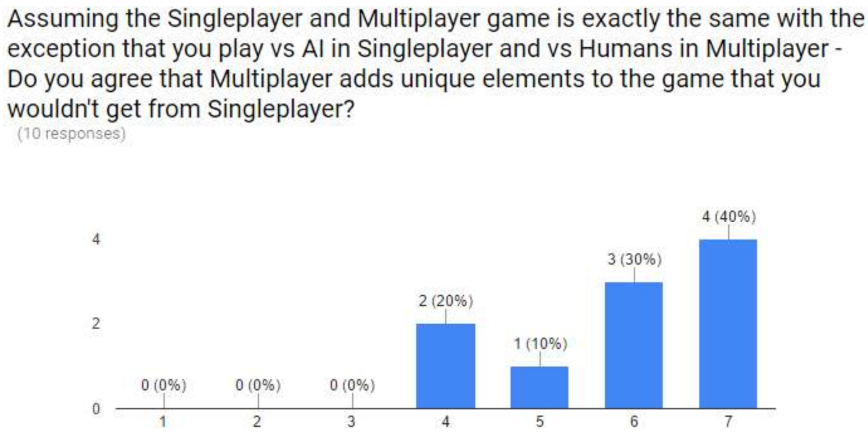

Figure 40 - ***Results for Unique Gameplay via Multiplayer***

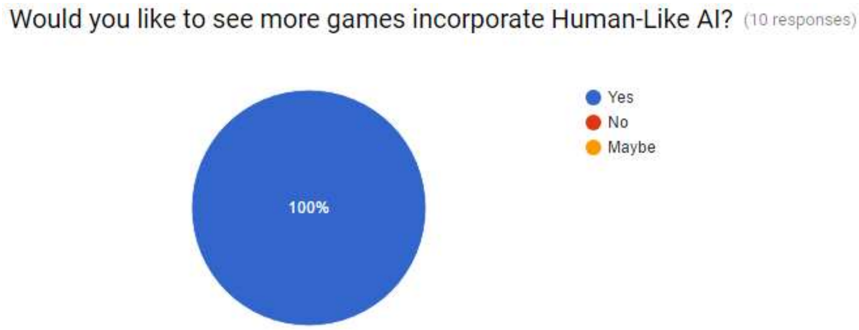

Figure 41 - ***Results for More Games with Human-Like AI***

These results prove very interesting and promising for the demand of the future of AI in video games and confirms previous statements on the state of Multiplayer in Real-Time Strategy (RTS) games seeing a large shift in popularity from Multiplayer or No Preference to Singleplayer only in Figures 38 and 39.

In addition to these opinion based questions, multiple included an additional text response for why the tester chose their answers. Testers often commented on Multiplayer in RTS games being overwhelming and not as relaxing as the Singleplayer alternative. The full results can be found in Appendix B.

4.2 Human vs AI

In order to test all AI techniques evenly testers were asked to perform 5 tests. Each test represented each of the developed AI. Testers had the choice of map, out of 5 available. Through internal testing it was confirmed that the map produced little to no difference in decision making so this was merely to allow testers to play a style of map they'd prefer. It is worth noting that all maps created for testing are symmetrical and even for all players (AI and Human) to ensure fair testing. From Figure 42 we can see a near even distribution across the available maps which is fairly fortunate as by pure chance. Testers were asked to select a single map to play through all 5 tests.

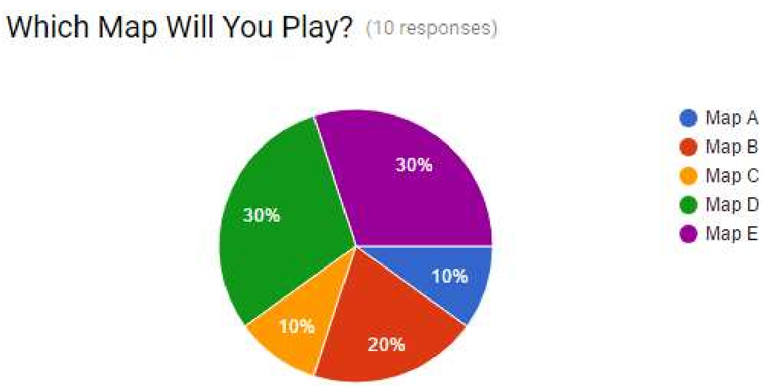

Figure 42 - *Results for Selected Map*

Each AI, excluding the "Simple" AIs, had randomly assigned skill levels upon loading the test with the two lowest skill levels excluded due to them being too weak. Each tester was also asked to note down the length of time they spent on each test in seconds as provided by the game when a winner is declared; this information is available in Appendix B.

Results in this section will always be in the same order as the test numbers presented to testers:

1. Simple Rule Base
2. Advanced Rule Base
3. Simple Fuzzy Logic
4. Advanced Fuzzy Logic
5. Probability Fuzzy Logic

To understand difference in AI types, see Section 3.2. Testers did not know which AI they were playing against.

4.2.1 Difficulty

Figures 43 to 52 display the percentage of testers who won or lost the match against the AI and then how the tester rated the challenge.

Simple Rule Base

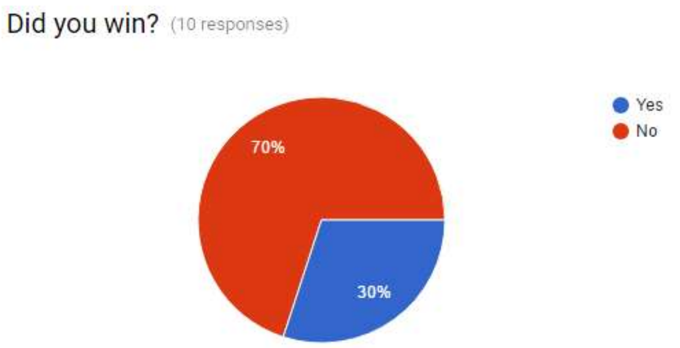

Figure 43 - *Results for Simple Rule Base Wins*

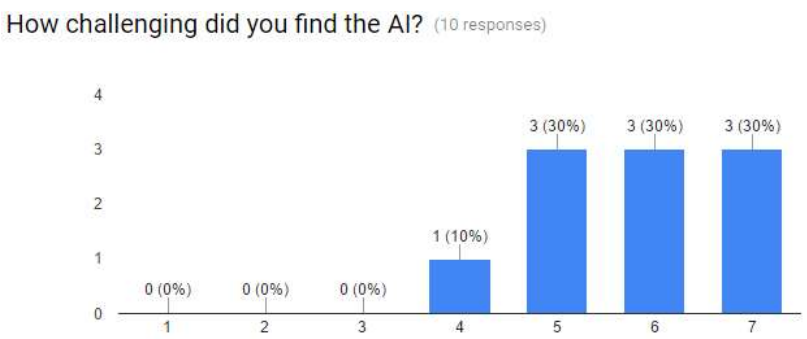

Figure 44 - *Results for Simple Rule Base Challenge*

Advanced Rule Base

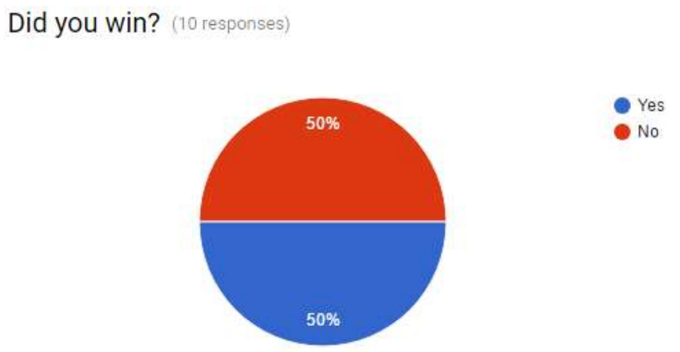

Figure 45 - *Results for Advanced Rule Base Wins*

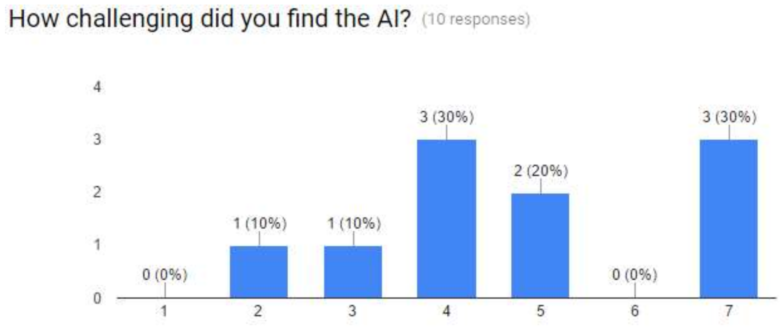

Figure 46 - *Results for Advanced Rule Base Challenge*

Simple Fuzzy Logic

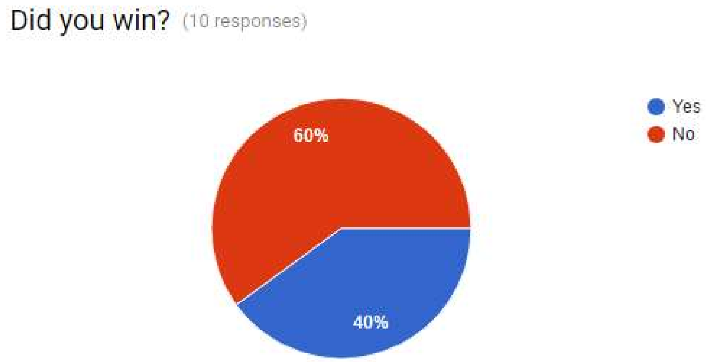

Figure 47 - *Results for Simple Fuzzy Logic Wins*

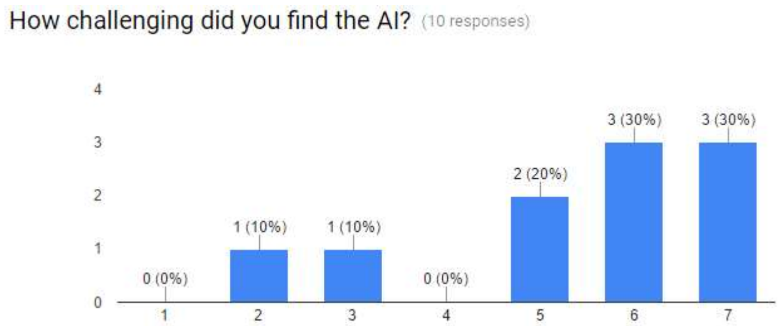

Figure 48 - *Results for Simple Fuzzy Logic Challenge*

Advanced Fuzzy Logic

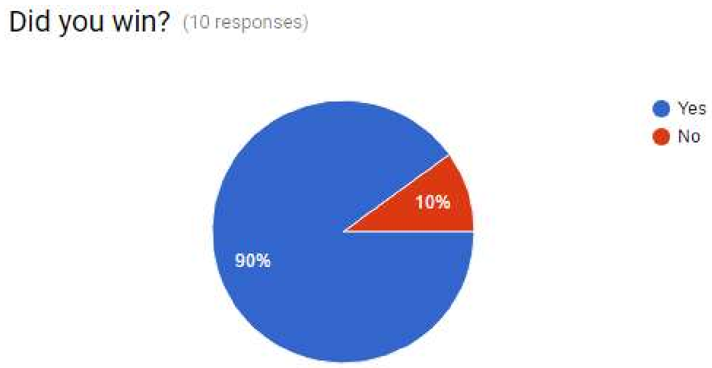

Figure 49 - *Results for Advanced Fuzzy Logic Wins*

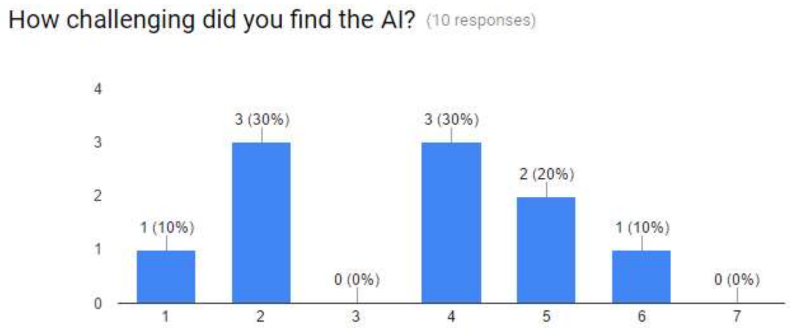

Figure 50 - *Results for Advanced Fuzzy Logic Challenge*

Probability Fuzzy Logic

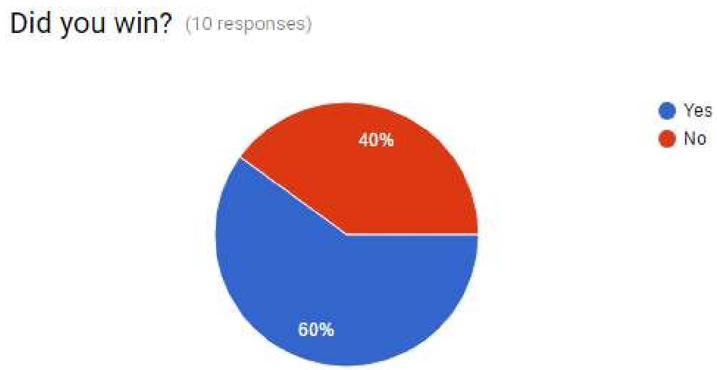

Figure 51 - *Results for Probability Fuzzy Logic Wins*

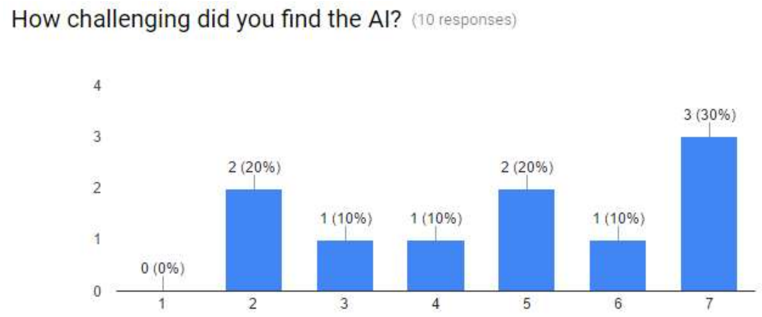

Figure 52 - *Results for Probability Fuzzy Logic Challenge*

These results follow the expected results for difficulty of AI techniques, clearly showing the Simple AIs being harder to beat; most likely due to them only following the results and having in-human reactions. Most surprising is the Advanced Fuzzy Logic being the easiest technique to beat; expected result would have been the Probability Fuzzy Logic due to its nature to make mistakes. It's unclear what in the Advanced Fuzzy Logic system could cause its win-rate to dramatically drop compared to other techniques without detailed analysis of its actions.

4.2.2 Realism

Figures 53 to 57 display the Human-Like feedback per AI technique from all testers.

Simple Rule Base

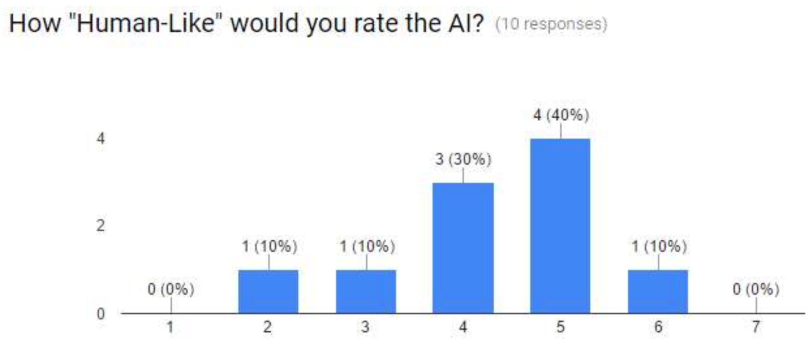

Figure 53 - *Results for Simple Rule Base Realism*

Advanced Rule Base

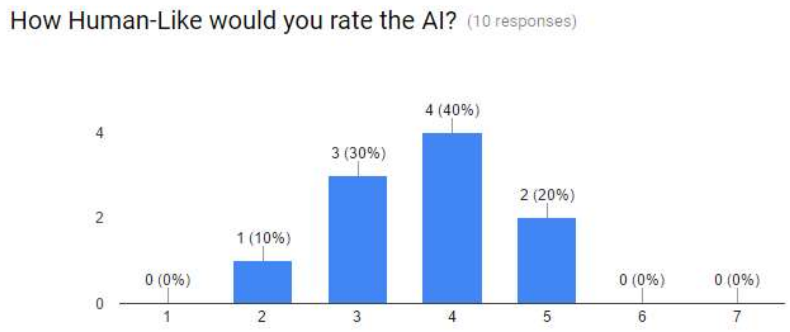

Figure 54 - *Results for Advanced Rule Base Realism*

Simple Fuzzy Logic

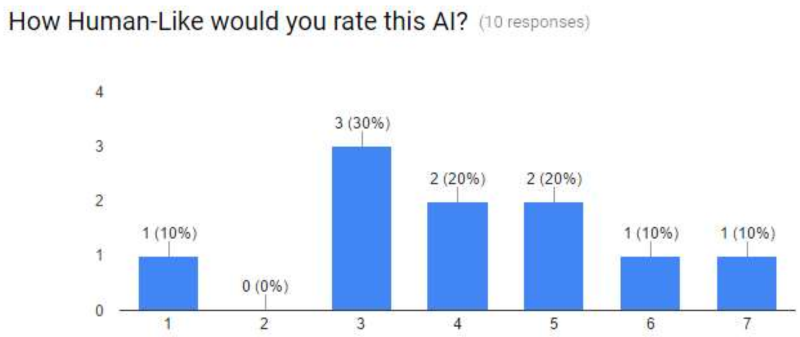

Figure 55 - *Results for Simple Fuzzy Logic Realism*

Advanced Fuzzy Logic

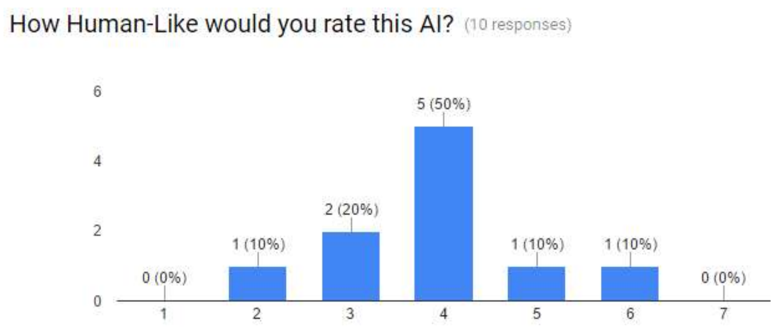

Figure 56 - *Results for Advanced Fuzzy Logic Realism*

Probability Fuzzy Logic

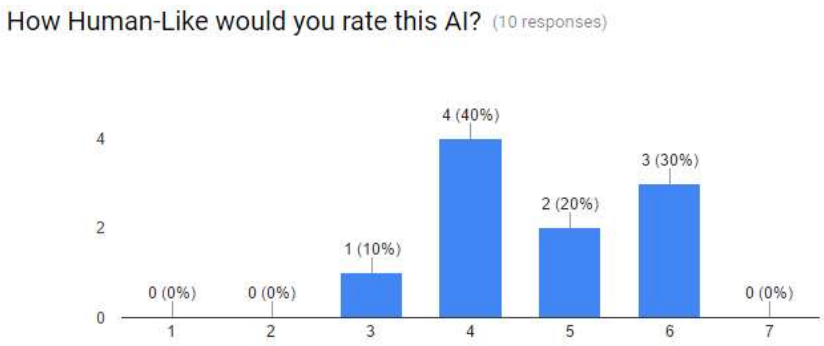

Figure 57 - *Results for Probability Fuzzy Logic Realism*

These results, although not as decisive as previous results still sway in the direction of expected results of AI realism increasing for the Advanced and Probability techniques. However, rather surprisingly, Simple Rule Base was rated closer to realism than the Advanced Rule Base; speculation of this result is testers swaying their answers based on difficulty while against the Simple AI techniques. This speculation will be later confirmed in interesting findings from the Turing Test.

4.2.3 Enjoyment

Figures 58 to 62 display tester rated enjoyment against each AI technique.

Simple Rule Base

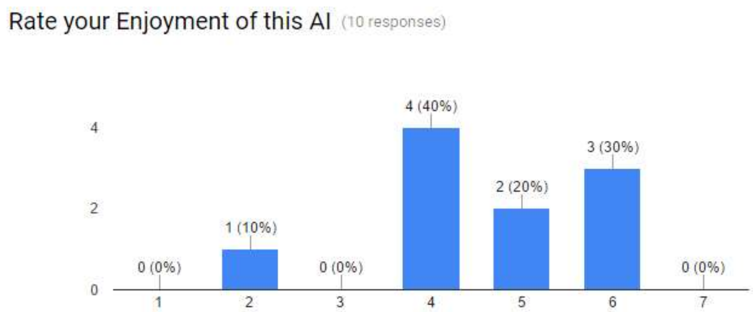

Figure 58 - *Results for Simple Rule Base Enjoyment*

Advanced Rule Base

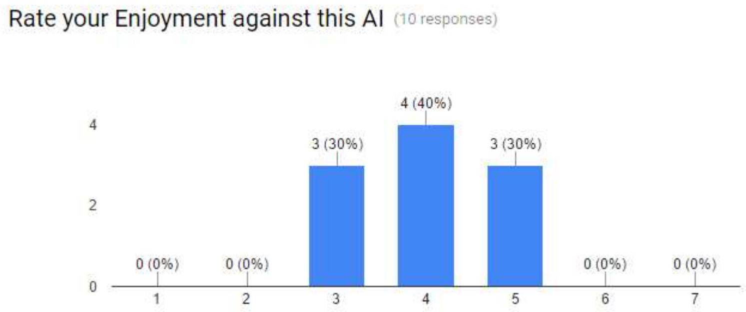

Figure 59 - *Results for Advanced Rule Base Enjoyment*

Simple Fuzzy Logic

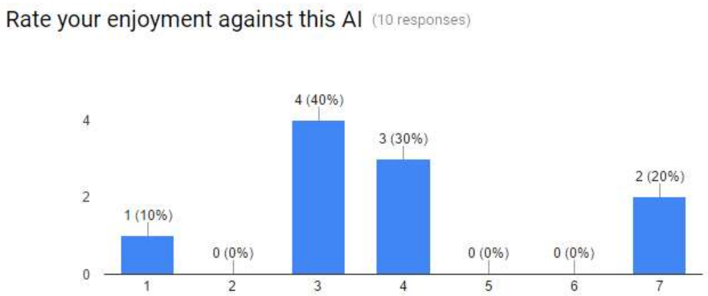

Figure 60 - *Results for Simple Fuzzy Logic Enjoyment*

Advanced Fuzzy Logic

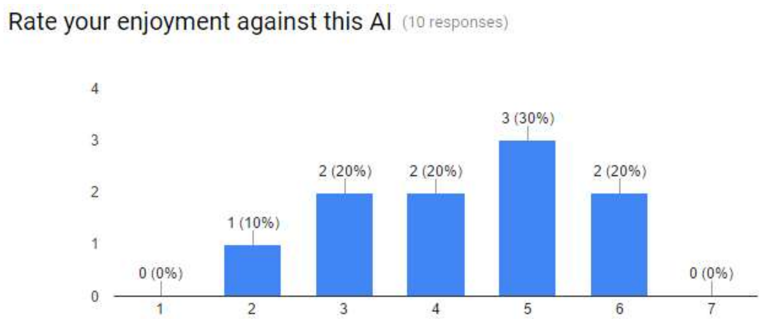

Figure 61 - *Results for Advanced Fuzzy Logic Enjoyment*

Probability Fuzzy Logic

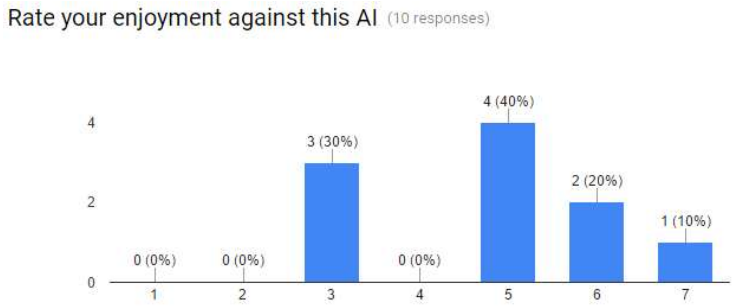

Figure 62 - *Results for Probability Fuzzy Logic Enjoyment*

These results without a doubt are the most pleasant, with a steady increase in enjoyment from Test 2 (Advanced Rule Base) to Test 5 (Probability Fuzzy Logic) with Test 1 (Simple Rule Base) again having its unique score. All of the results of this section clearly show each AI technique was successful in developing a steady increase in all factors as tests progressed. It's interesting to note that the obscure results of the Simple Rule Set AI is likely due to the AI being familiar with players and easier to understand.

After each test testers were able to comment on the AI they played against, full comments can be found in Appendix B; however, here are some highlight observations include:

- The Simple Rule Base AI was fast to react and only made short term decisions.
- Advanced AI did not feel any different from Test 1 (Simple Rule Base)
- Simple Fuzzy Logic appeared very aggressive.
- Advanced Fuzzy Logic was defensive and slower.
- One tester noted the Probability Fuzzy Logic made human-like mistakes.

4.3 AI vs AI

To evaluate the performance of AI against each other an idea inspired by the Civilization game series (Firaxis Games 1991) is “AI Battle Royale” a game mode that, in Civilization, lasts an extremely long time and pits lots of AI against each other with the intent of watching the AIs decision making over long periods of time and watch the world map in the game change as factions grow and fall. A few problems exist with this idea that were adjusted for the implementation of it in Little Planets. Speed; AI Battle Royales are famous for lasting months and some years; this is obviously not doable for an honours project so the AI will play at 5 times regular speed (with regular speed being around twice as fast as intended in-order to keep games short). Uneven maps; in Civilization the AI face each other most commonly on the Earth world map; this provides an uneven playing field for AI placed in harder to populate areas. This test attempts to be as fair as possible by placing AIs on a symmetrical map with the exact same starting situation. Figure 63 displays the results of 15 AI battles across 3 perfectly symmetrical maps.

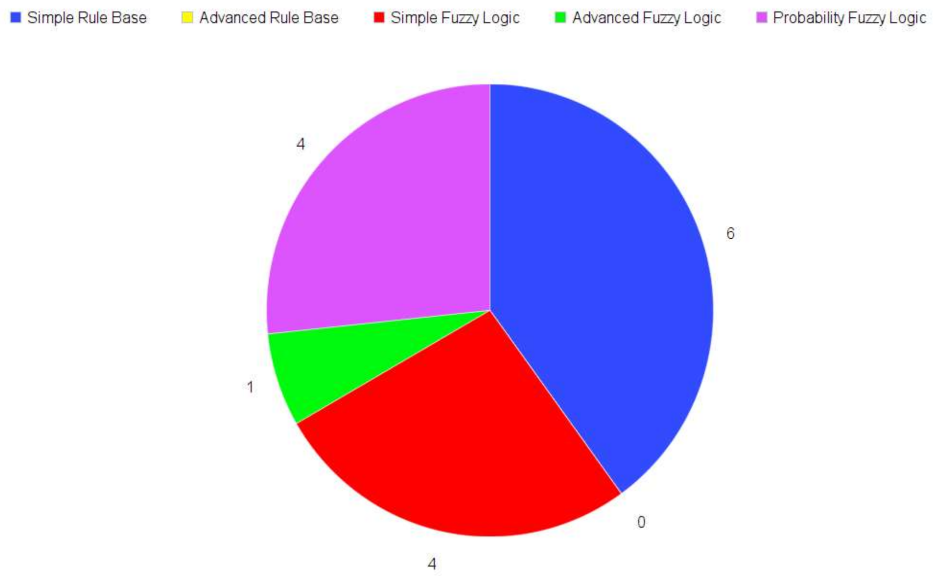

Figure 63 - *Results for Winners of AI Brawls*

It is slightly surprising with Probability Fuzzy Logic having win-rate that matches Simple Fuzzy Logic as the expected results would be for the highest win-rates to belong to both Simple system as they do not have reaction times and simply take what they want rather than take into consideration politics of the game.

4.4 The Turing Test

The ultimate test of whether an AI is human-like is the Turing Test. Unfortunately, as previously mentioned, implementation of an actual multiplayer mode would not have been possible. Instead this Turing Test told testers they were about to take part in the Turing Test. Testers were then misled to believe they were connecting to a multiplayer game. Once connected the tester would play against the same AI (one of each) 3 times on 3 different maps with no human player. Their task was to figure out which faction was human; thus telling which AI was the most human-like. All testers were successfully misled into believing they were connected to a multiplayer session. Figure 64 displays the results of the test, the colours that the AIs controlled is the same colours associated with them in Figure 64.

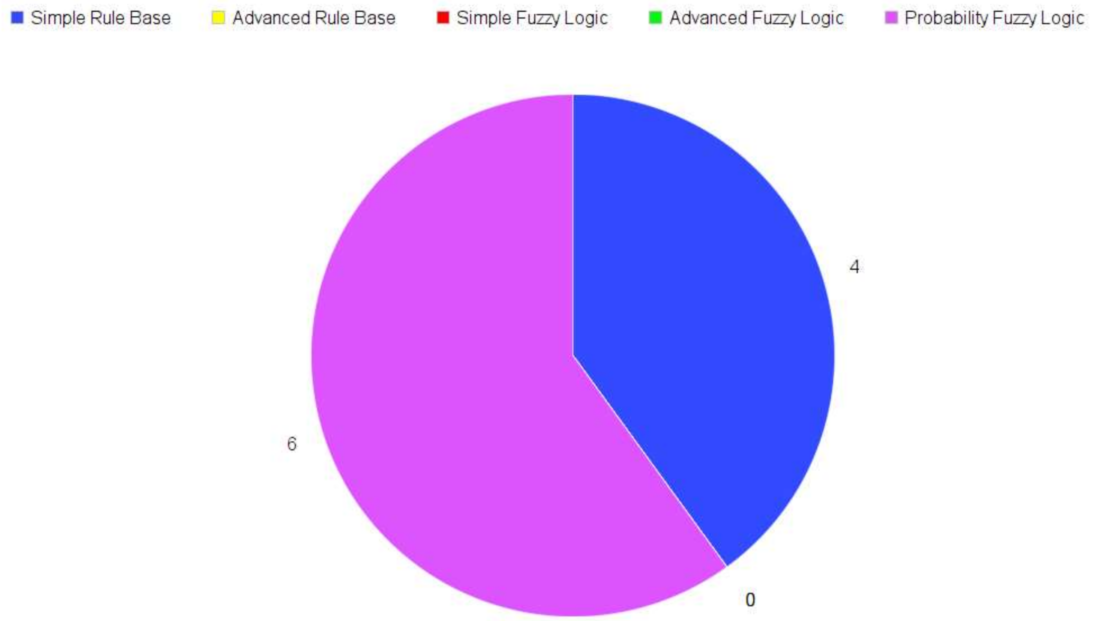

Figure 64 - *Results for the Turing Test*

In addition to selecting their chosen answer, testers were asked to rate their confidence of the answer and comment on why they believe this to be the case. Figure 65 displays the confidence results.

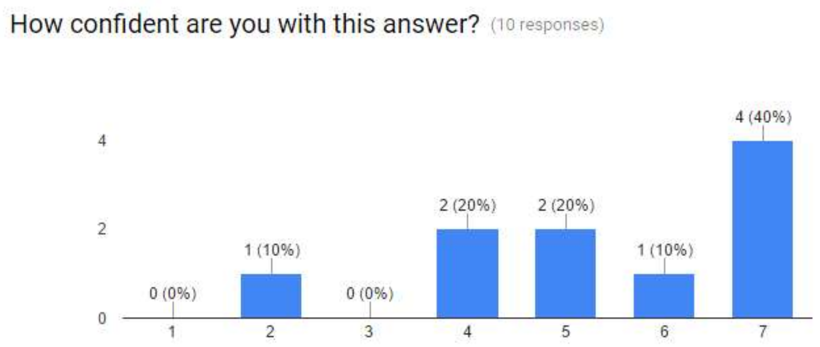

Figure 65 - ***Results for the Turing Test Confidence***

Interestingly, all testers who selected 7 on the confidence scale selected Pink (Probability Fuzzy Logic) as their answer. These results are remarkable. Probability Fuzzy Logic is a clear winner on human-likeness successfully fully convincing 4 testers that it was human. An even more interesting observation is the comments left by those who selected Dark Blue (Simple Rule Base) as their answer: two of the four who selected Dark Blue said that they initially thought Pink but due to Pink making silly actions (mistakes) they changed their answer to Dark Blue.

However, by far the most interesting observation was made by one of the testers who marked themselves as being extremely well engaged in the conversation in AI. They their comment is as follows:

“[pink] was very strong initially and then seemed to play stupid for a while at one point he just somehow managed to spawn 5000 units with no planets showing that he had that many so i believe he was hiding them in the planets that are too small to see so it was apparent that an AI would take a huge amount of programming to do this” (Appendix B, BC 4)

This tactic is definitely not programmed into the AI but rather somehow the Probability AI continuously rolled to keep those units stationary in the planet despite the odds being extremely low of this happening it managed to happen. It was not until the tester reached the planet that the probability rolled to move the units out of the planet, surprising the tester and winning the game for the AI.

5 Discussion

This section will summarise and discussion of results in Section 4 with a focus on how results relate to the research question and aims of the project. Including discussion of the limitations and concerns throughout the project execution and testing.

5.1 General Findings

The project found that the decision of which AI technique to implement with matters in regards to human-likeness. The human-likeness of an AI increases even greater when various human-like qualities are applied to it such as the reaction times and trust mechanic introduced in this project. This project also found that there is high demand for a more human-like AI and through testing it is confirmed to improve the user's experience.

5.2 Human-Like Qualities

The human-like qualities proposed and implemented for this project proved to be of a greater importance to the player experience and human-likeness of the tested AI that first anticipated. Had this project had more time it is a firm believe that further implementation of human-like qualities such as AI personalities would enhance the player experience by a great deal with every quality implemented.

Although the diplomacy mechanic was not liked or utilised by all testers, they did take notice to AI making decisions based on the trust they had towards a faction and feedback on this feature is greatly positive.

5.2.1 AI Techniques

This project has proven different AI techniques make a significant difference when attempting to mimic human player decision making. A steady increase in human-likeness is seen from Rule Base systems to Fuzzy Logic which peaks with the Probability Fuzzy Logic. This is reinforced by the results of the Turing Test which had 6 of 10 testers believe that the Probability AI was an actual human player.

By far the most interesting finding however is hidden behind the success of the Turing Test results. While the Probability AI was rated the most human-like the only other AI to gain votes was the Simple Rule Based AI, an AI that has repeatedly produced obscure results outwith the smooth flow between tests. Most comments on this commented on the Simple Rule Base being the strongest and winning by making the right moves. In addition to this, no faction that consistently lost was ever selected as most human-like. This finding speaks volumes on the current state of AI in games, that an AI can gain a vote for being human-like based on it being the strongest. Suggesting that the general consensus is that humans are always better than the AI counterparts at playing video games.

5.2.2 In A Real-Time Strategy Game

During the development of this project, it became more and more clear why real-time strategy (RTS) games often avoid more complex AIs with the introduction of human-like qualities. The amount of actions and rules that need to be considered while developing AI for an RTS is very high; more time was spent thinking of how to combat a problem than typing code since so many things must be taken into consideration. Testing and balancing is also a real issue for RTS games as this also became apparent with the development of the AI systems for this project; either a lot of testers must be hired or developers must spend countless hours playing their games to find every small bug and unexpected action that an AI can make. These things increase dramatically as projects get bigger. However, with proper implementation, testing and management of decision making RTS games still have a lot of room for improvement and fixing these issues is the first step in order to get more human-like AI into RTS games.

The implementation of the human-like qualities in the RTS environment was met with extremely positive feedback which hopefully proves and inspires developers to push to develop an AI that can match the experience currently only obtainable from multiplayer modes in strategy games.

5.3 The Player Experience

One of the most defined improvements from all test data is the gradual improvement in enjoyment as the tests progressed. Probability AI was rated the most enjoyable AI to play against. Everyone is different so to gauge what affects this decision is extremely hard if at all possible. however, it is clear from these findings that the human-like qualities and differences in AI techniques are to credit for the increase enjoyment during these tests. As AI were very closely related with the only additions of the human-like qualities, such as reaction times, and how the decisions were made.

5.4 Concerns and Limitations

The primary concern through the testing phase of this project was the issue that RTS games of this nature are not meant to be fast paced and instead meant to be played over long periods of time. This is obviously impractical for this project as it is unreasonable to ask testers to spend hours testing each AI. Were it possible it would give more concrete evidence that these decision making techniques and human-like qualities can positively affect a true real-time strategy game.

Another concern is the addition of game features greatly increases work required, with every feature implemented you must think how each AI would utilize the feature, all the possible ways it can be used and how it affects each other feature within the game. If anymore game features were added to this game during the duration of the honours research period of the project it would be unlikely that the game were to be complete as with this many features although small on paper takes an incredible amount of code to create and maintain. Throughout the development of the game environment major bug reports that were not resolved immediately were stored in a spreadsheet which can be found in Appendix D.

6 Conclusion

The success of the project is primarily based on whether the research question was investigated and answered. The research question was:

Which AI techniques are more effective at simulating a human-player's actions in a Real-Time Strategy game and how does having an AI with human-like qualities affect the player experience?

The question can be broken down and answered in several parts. Using two core AI Techniques: Rule Base and Fuzzy Logic a total of 5 techniques were created and tested. A "Simple" variant; which is the standard approach taken by each technique wherein there is no human-like qualities other than those that can be thought to be from the decision making process each takes. An "Advanced" variant which utilises the human-like qualities implemented for this project. Finally a Probability solution which takes the Advanced Fuzzy Logic system and removes the fuzzy output, replacing output with a randomly selected value which dictates the decision based on the influence values which would have been used for the fuzzy calculation. The Probability system proved to be the most effective at simulating human-player actions by not only being the most highly rated for human-likeness but also passing a custom version of the Turing Test for this project in 60% of tests. Simple Rule Base was surprisingly selected as the second most human-like mostly due to testers selecting it because it won; which provides excellent commentary on the current state of AI in Real-Time Strategy games.

Having an AI with human-like qualities has proven, through testing conducted in this project, to greatly increase enjoyment while sustaining a high rating on how challenging the experience was.

6.1 Future Work

This project set out with an incredibly large scope which was drawn in as the year entered 2017. Originally the project sought to investigate neural networks and a theorized neuro-fuzzy system by Ertürk (Ertürk 2009); however, due to time constraints these techniques were not able to be brought further in this project. The game environment and its code framework has been created in such a way that allows for smooth and relatively easy inclusion of additional techniques in the AI Controller script. Through the research conducted and the fuzzy logic systems implemented the project can also back Ertürk theory (Ertürk 2009) of a neuro-fuzzy system that allows fuzzy logic to control short term decisions and neural networks for long term decisions - this would perhaps be the most interesting thing were anyone to continue on with this project or conduct similar research.

Late into development of the project developers of Stellaris (Paradox Interactive 2016) released a video (Bari, M. A 2017) giving details of how their AI system works, revealing that they use data driven design to essentially give AIs personalities when a game starts. Data driven design was something never considered for this project but now with this knowledge if this project were to be redone there would be a much greater focus at the implementation of human-like qualities starting by the assignment of personalities and how they affect decision making through the use of data driven design.

When this project was conceived it was due to the reason that there were no cell-based games that offered mechanics such as diplomacy or remotely human-like AI. The genre is filled with standard rule-based systems which hold the otherwise great genre back. This project was not only created as research project but also to inspire other developers to be more ambitious with the creation of their AI and finally hopefully have the game developed into a full game that is able to showcase AI techniques and how much of a difference adding human-like qualities makes to real-time strategy games.

Finally, as brought up repeatedly throughout this dissertation; balancing is a huge issue for RTS games and this project is not an exception. There are many balancing issues which given the proper time and treatment when fixed would provide even greater results as balancing plays a large part in human and AI decision making in RTS games.

Appendices

All Appendices are available in the execution submission or web links below.

A - Questionnaire

Available from: http://duncanbunting.com/honours/questionnaire.pdf [Accessed 24 April 2016]

B - Questionnaire Responses

Available from:

https://docs.google.com/spreadsheets/d/16irvy5TADXyjVQV3ESAu4mccM9UaAtTJkEtBAx1R94w/edit#gid=24241568 [Accessed 24 April 2016]

C - APM / Reaction Time Data

Available from:

https://docs.google.com/spreadsheets/d/18hOdvChjevfQ9Kwy6YAiQcAlQLwZdgU56aaE0qrRisQ/edit?usp=sharing [Accessed 24 April 2016]

D - Bug Report Spreadsheet

Available from:

https://docs.google.com/spreadsheets/d/12p0X1gzYBFUoG_e5MWOHzeraInjya281B3AQw_sOzAE/edit?usp=sharing [Accessed 24 April 2016]

References

Alien: Isolation. 2014. [computer game]. Creative Assembly.

Anno 2070. 2011. [computer game]. Blue Byte.

Anon. 2016a. *Ex Machina- Ava wasn't a real AI*. Available from: https://www.reddit.com/r/FanTheories/comments/3igoc3/ex_machina_ava_wasnt_a_real_ai/ [Accessed 13 Feburary 2017].

Anon. 2016b. *[Ex Machina] The movie is a Turing test for the audience.* Available from: https://www.reddit.com/r/FanTheories/comments/3q0c5y/ex_machina_the_movie_is_a_turing_test_for_the/ [Accessed 13 Feburary 2017].

Anon. 2016c. *The AI is absolutely suicidal*. Available from: https://www.reddit.com/r/hoi4/comments/4n42lr/the_ai_is_absolutely_suicidal/ [Accessed 7 January 2017].

Anon. 2013. AI Building - Will Ubisoft Improve It EVER?. Available from: http://steamcommunity.com/app/48240/discussions/0/666825525084910018/ [Accessed 7 January 2017].

Bari, M. A. 2017. *Creating complex AI behaviour in Stellaris through Data Driven Design*. [video]. GDC. Available from: http://gdcvault.com/play/1024223/Creating-Complex-AI-Behavior-in [Accessed 14 March 2017].

Bourg D.M., Seemann,G. 2004. *AI for Game Developers*. 1st ed. O'Reilly Media

Colby, K. 1972. *PARRY: The A.I. chatterbot from 1972.* Available from: https://phrasee.co/parry-the-a-i-chatterbot-from-1972/ [Accessed 10 Feburary 2017].

Civilization. 1991. [computer game series]. PC. Firaxis Games.

Civilization V. 2010. [computer game]. PC. Firaxis Games.

Creatures. 1996. [computer game series]. Millennium Interactive.

Dune II. 1992. [computer game]. Westwood Studios.

Electronic Engineering. 1951. *Nimrod*. [online image]. Available from: https://www.wired.com/2010/06/replay/ [Accessed 03 April 2017].

Ertürk, U.R. 2009. *Short Term Decision Making with Fuzzy Logic And Long Term Decision Making with Neural Networks In Real-Time Strategy Games*. Available from: http://www.hevi.info/tag/artificial-intelligence-in-real-time-strategy-games/ [Accessed 01 October 2016].

EVER Starleague. [no date]. *Highest Ever Recorded APM*. [online image]. Available from: http://pds7.egloos.com/pds/200805/10/51/f0038051_48258706320d1.png [Accessed 03 March 2017].

Ex Machina. 2015. [film]. Directed by Alex Garland. Film4, DNA Films.

Fullér, R. 1995. *Neural Fuzzy Systems*. Available from: http://uni-obuda.hu/users/fuller.robert/ln1.pdf [Accessed 19 October 2016].

Galcon. 2008. [computer game]. Phil Hassey.

GameSpot. 2016. *Stellaris Screenshot*. [online image]. Available from: https://www.gamespot.com/reviews/stellaris-review/1900-6416431/ [Accessed 20 April 2017].

Grant, E., Lardner, R. It. The Talk Of The Town. *The New Yorker.* 2 August: p. 18. Available from: http://www.newyorker.com/magazine/1952/08/02/it [Accessed 03 April 2017].

Gun Fight. 1975. [computer game]. Taito Corporation.

Hardawar, D. 2015. *'Ex Machina' shows Turing isn't enough to test AI*. Available from: https://www.engadget.com/2015/04/10/ex-machina-review/ [Accessed 13 Feburary 2017].

Hearts of Iron 4. 2016. [computer game]. Paradox Interactive.

Hello Neighbour. 2017. [computer game]. tinyBuild.

Hruska, J. 2015. *Why AI still sucks in video games - and how DirectX 12 could change it for the better*. Available from: https://www.extremetech.com/gaming/219358-why-ai-still-sucks-in-video-games-and-how-directx-12-could-change-it-for-the-better [Accessed 26 September 2016].

Human Benchmark. 2017. Available from: http://www.humanbenchmark.com/ [Accessed 03 March 2017]

Jensen K. 2016. *What happened to the real-time strategy genre?*. Available from: http://www.geek.com/news/what-happened-to-the-real-time-strategy-genre-1649869/ [Accessed 21 September 2016].

Kosta. 1992. *Dune II Screenshot*. [online image]. Available from: http://www.abandonia.com/en/games/36/Dune+II+-+The+Building+of+a+Dynasty.html [Accessed 20 April 2017].

Little Stars for Little Wars 2. 2012. [computer game]. Android. MKG.

Loebner Prize. 2015. *Home Page of The Loebner Prize in Artificial Intelligence "The First Turing Test"*. Available from: http://www.loebner.net/Prizef/loebner-prize.html [Accessed 14 March 2017].

Melis, G. 2011. *Google AI Challenge: Planet Wars*. Available from: http://franz.com/services/conferences_seminars/webinar_1-20-11.gm.pdf [Accessed 10 September 2016].

Nimrod. 1951. [computer hardware]. Ferranti.

Pelletier, F. 2000. *Metamathematics of Fuzzy Logic*. Available from: http://www.sfu.ca/~jeffpell/papers/ReviewHajek.pdf [Accessed 29 March 2017].

Rowley, J. 2017. *Hearts of Iron 4 AI and Realism*. Available from: https://steamcommunity.com/app/394360/discussions/0/144512875332398805/ [Accessed 6 January 2017].

Speed Race. 1974. [computer game]. Taito Corporation.

Starcraft. 1998. [computer game series]. PC. Blizzard Entertainment.

Starcraft II. 2010. [computer game]. PC. Blizzard Entertainment.

Starcraft Wiki. 2012. *Actions Per Minute Scale*. Available from: http://starcraft.wikia.com/wiki/Actions_per_minute [Accessed 02 March 2017].

Stellaris. 2016. [computer game]. Paradox Interactive.

Super-Soviet. 2015. *Alpha Ethics System in Stellaris*. [online image]. Available from: https://forum.paradoxplaza.com/forum/index.php?threads/the-ethics-of-famous-sci-fi-races.887596/ [Accessed 20 April 2017].

The Turing Test. 2016. [computer game]. Bulkhead Interactive.

Total War. 2000. [computer game series]. PC. Creative Assembly.

Total War: Rome II. 2014. [computer game]. PC. Creative Assembly.

Turing, A. 1950. *Computing Machinery and Intelligence*. Available from: http://loebner.net/Prizef/TuringArticle.html [Accessed 14 October 2016].

Turing, A. [no date]. Quoted in: Guardian Northerner. 2012. *Alan Turing: "I am building a brain." Half a century later, its successor beat Kasparov*. Available from: https://www.theguardian.com/uk/the-northerner/2012/may/14/alan-turing-gary-kasparov-computer [Accessed 15 October 2016].

Unity Technologies. 2005. *Unity*. [software]. Available from: https://unity3d.com/ [Accessed 01 January 2017].

Walker, J. 2011. *Look, Single-Player People Are Just Better*. Available from: https://www.rockpapershotgun.com/2011/08/01/look-single-player-people-are-just-better/ [Accessed 30 September 2016].

Weizenbaum, J. 1966. *ELIZA, computer therapist.* [program]. Available from: http://www.chayden.net/eliza/Eliza.html [Accessed 19 March 2017].

Zadeh, L.A. 1965. *Fuzzy Sets*. Available from: https://people.eecs.berkeley.edu/~zadeh/papers/Fuzzy%20Sets-1965.pdf [Accessed 01 October 2016].

Bibliography

Anon. 2015. *Multiplayer mind-games and scheming?*. Available from: https://www.reddit.com/r/civ/comments/31msmv/multiplayer_mindgames_and_scheming/ [Accessed 27 September 2016].

Anon. 2010. *A Short Fuzzy Logic Tutorial*. Available from: http://cs.bilkent.edu.tr/~zeynep/files/short_fuzzy_logic_tutorial.pdf [Accessed 09 January 2017].

Bengio, Y. 2012. *Why aren't artificial neural networks used for everything?*. Available from: https://www.quora.com/Why-arent-artificial-neural-networks-used-for-everything [Accessed 16 October 2017].

Bourg, D.M., Seemann, G. 2004. *Four Cool Ways to Use Neural Networks in Games*. Available from: http://www.onlamp.com/pub/a/onlamp/2004/09/30/AIforGameDev.html [Accessed 16 October 2016].

Burford, G. B. 2016. *Alien Isolation's Artificial Intelligence Was Good… Too Good*. Available from: http://kotaku.com/alien-isolations-artificial-intelligence-was-good-too-1714227179 [Accessed 20 October 2016].

Buro, M. 2004. *Call for AI Research in RTS Games*. Available from: https://skatgame.net/mburo/ps/RTS-AAAI04.pdf [Accessed 21 September 2016].

CNET. 2015. *What is the Turing Test?*. Available from: https://www.youtube.com/watch?v=sXx-PpEBR7k [Accessed 21 Feburary 2017].

D'Anastasio, C. 2016. *A Horror Game Where The AI Learns From Your Every Move*. Available from: http://kotaku.com/a-horror-game-where-the-ai-learns-from-your-every-move-1787241885 [Accessed 30 October 2016].

Delahunty-Light, Z. 2016. *Are you the villain in Hello Neighbor? The AI thinks so and learns your tactics to hunt you down*. Available from: http://www.gamesradar.com/are-you-the-villain-in-hello-neighbor-the-ai-thinks-so-and-learns-y our-tactics-to-hunt-you-down/ [Accessed 30 October 2017]

Dyer, M. 2013. *Stardock's Galactic Civilizations 3 Revealed*. http://uk.ign.com/articles/2013/10/15/stardocks-galactic-civilizations-3-revealed [Accessed 03 November 2016].

Elliot, J. 2015. *Moving Dots Around. The early years of Video Games.* Available from: https://www.slideshare.net/JasonElliott8/history-of-games-class-1-moving-dots-around-48248853 [Accessed 04 April 2017]

Foster, S. [no date]. *Fuzzy Logic*. Available from: http://www.fcet.staffs.ac.uk/rgh1/cai/foster/Fuzzy Logic.ppt [Accessed 03 October 2016].

Fullér, R. 2001. *Neuro-Fuzzy Methods*. Available from: http://uni-obuda.hu/users/fuller.robert/dam.pdf [Accessed 20 October 2016].

Gendler, A. 2016. *The Turing Test: Can a computer pass for a human?*. Available from: https://www.youtube.com/watch?v=3wLqsRLvV-c [Accessed 25 March 2017].

Makedonski, B. 2016. *EVE Online developer has the right attitude about making sandbox games*. Available from: https://www.destructoid.com/eve-online-developer-has-the-right-attitude-about-making-sandbox-games-358211.phtml [Accessed 30 September 2016].

Metz, C. 2016. *What The AI Behind AlphaGo Can Teach Us About Being Human*. Available at: https://www.wired.com/2016/05/google-alpha-go-ai/ [Accessed 03 November 2016].

McNeill, E. 2011. *A New Artificial Intelligence for Auralux*. Available from: http://www.cs.dartmouth.edu/reports/TR2011-692.pdf [Accessed 01 September 2014].

Millington, I. 2006. *Artificial Intelligence for Games*. 1st ed. Elsevier.

Ontañón, S., et al. 2015. *RTS AI: Problems and Techniques*. Available from: https://webdocs.cs.ualberta.ca/~cdavid/pdf/ecgg15_chapter-rts_ai.pdf [Accessed 03 November 2016].

Patel, P. 2011. *Fuzzy Logic for Games*. Available from: http://purvag.com/blog/?p=284 [Accessed 08 October 2016].

Rabin, S. 2008. *AI Game Programming Wisdom 4*. 1st ed. Charles River Media.

de Reus, N. M. 1994. *Assessment of benefits and drawbacks of using fuzzy logic, especially in fire control systems*. Available from: http://www.dtic.mil/dtic/tr/fulltext/u2/a285428.pdf [Accessed 08 October 2016].

Robertson, G, Watson, I. 2014. *A Review of Real-Time Strategy Game AI*. Available from: http://www.aaai.org/ojs/index.php/aimagazine/article/view/2478 [Accessed 03 November 2016].

Keylary. 2008. *Introduction to C# and Fuzzy Logic*. Available from: https://www.codeproject.com/Articles/23387/Introduction-to-C-and-Fuzzy-Logic [Accessed 08 Feburary 2017].

Macri, D. 2011. *An Introduction to Neural Networks with an Application to Games*. Available from: https://software.intel.com/en-us/articles/an-introduction-to-neural-networks-with-an-application-to-games [Accessed 27 Feburary 2017].

Scott, B. 2003. *AI Game Programming Wisdom 2*. 1st ed. Charles River Media.

Schwab, B. 2004. *AI Game Engine Programming*. 1st ed. Charles River Media.

See, J. 2016. *Game AI: Fuzzy Logic*. Available from: http://pesona.mmu.edu.my/~johnsee/teaching/gamealgo/material/GameAI_FuzzyLogic.pptx [Accessed 02 October 2016].

Stene, S.B. 2006. *Artificial Intelligence Techniques in Real-Time Strategy Games - Architecture and Combat Behaviour*. Available from: http://www.diva-portal.org/smash/get/diva2:348419/FULLTEXT01.pdf [Accessed 01 November 2016].

Walther, A. 2006. *AI for real-time strategy games*. Available from: https://www.itu.dk/image/edu/theses/pdf/AndersWalther06.pdf [Accessed 01 November 2016].

Weber, B.G., Mateas, M., Jhala, A. 2011. *Building Human-Level AI for Real-Time Strategy Games*. Available from: https://games.soe.ucsc.edu/sites/default/files/weber-acs2011.pdf [Accessed 03 November 2016].

Welch, E. 2007. *Designing AI Algorithms For Turn-Based Strategy Games*. Available from: http://www.gamasutra.com/view/feature/129959/designing_ai_algorithms_for_.php [Accessed 20 September 2016].

Zadeh, L.A. 1979. *Fuzzy Sets And Information Granularity*. Available from: https://people.eecs.berkeley.edu/~zadeh/papers/Fuzzy%20sets%20and%20information%20granularity%201979%20Zadeh.pdf [Accessed 05 October 2016].