Applying Machine Learning Agents to enhance Player Experience in Video Games

Eman Paul Abela

Supervisor: Dr Owen Sacco

June - 2023

A dissertation submitted to the Institute of Information and Communication Technology in partial fulfilment of the requirements for the degree of BSc (Hons) Software Development

Authorship Statement

This dissertation is based on the results of research carried out by myself, is my own composition, and has not been previously presented for any other certified or uncertified qualification.

The research was carried out under the supervision of Dr Owen Sacco - Applying Machine Learning Agents to enhance Player Experience in Video Games, Eman Paul Abela

. . . .

Date Signature

Copyright Statement

. . . .

Date   Signature

Abstract

Video games are one of the most popular sources of entertainment in this day and age. With artificial intelligence being applied in a wide variety of areas, it is only a matter of time before AI revolutionises the video game industry as well. NPCs nowadays mostly use a simple logic state which makes them quite predictable. Therefore what if enemies were allowed to have minds of their own? Where each enemy would know what to do by itself? With the use of machine learning, this is possible as it allows agents to think freely and use the environment around them with their actions and ultimately provide an engaging experience to the player. However, engagement goes hand in hand with the challenge provided to the player. Therefore dynamic difficulty would also allow the player to have a customised experience for themselves where the agents are as strong or as weak as is required for them. This study aims to investigate the possibility of increasing player experience with the application of machine learning agents in NPCs. This will be tested on the general public by letting them play a dynamically difficult game made with Unity. The dynamic difficulty will be handled by a simple script which customises the environment and the stats and the in-game adversaries will be controlled by a neural network developed using Unity's ML-Agents Toolkit. Afterwards, the participants filled in a short questionnaire which allowed the compilation of ratings for the difficulty, adversaries' believability and the overall engagement experience. Overall the results showed that both the difficulty and the agent's believability required further improvement. However, the overall player engagement showed that the game itself was still immersive enough for most users. Overall the study finds that the prototype developed did provide an engaging experience to the player however the agents themselves still require further improvements before being applied to commercial games.

Keywords: Unity, Unity ML-Agents, Video Games, Neural Networks, Dynamic Difficulty, Immersion with Video Games, Player Experience

Acknowledgements

Firstly, I would like to thank my mentor Dr. Owen Sacco, who without his support, none of this would have been possible. The constant support and guidance were truly appreciated.

Secondly, I would also like to thank Mr. Darren Scerri who offered me general guidance in order to meet my deadlines.

Thanks also to my mother Anne Marie, my father Richard Jude and my brother Dylan Luke who helped keep me focused and kept pushing me forward to do all that was necessary to complete my dissertation.

I would like to also thank my friend Simona whose constant support and checking in allowed me to finish my dissertation and helped me keep pushing forward when things were looking bad. I would also like to thank her for proofreading my work.

I would lastly like to thank my friends Aidan, Natalia, Louie, Simona, Josianne and my family for sharing my prototype and survey in order to gather the results.

Table of Contents

Acronyms

NN	Neural Network
ML	Machine Learning
DL	Deep Learning
DCNN	Deep Convolutional Neural Network
FSM	Finite State Machine
DDA	Dynamic Difficulty Adjustment
NPC	Non-Playable Character
FPS	First Person Shooter

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

This study is motivated by the belief that games' difficulty should be subjective to the skills of the player instead of preset difficulty levels which may be too easy or too hard for the player. Using Machine Learning (ML) and deep learning, this can be achieved by switching out simple logic non-player characters (NPCs) currently have with artificial intelligence (AI), and having them act as enemies against the player. Usually, NPC enemies operate on a set of logical statements such as following a patrol path, if the player is found then following the player, if within attack distance, then attacking. However, this seems to be quite unrealistic in certain cases especially when difficulty is lowered and the enemies are made to act slow. Machine learning can be trained to have complex behaviour with unpredictable actions to have a more realistic and organic relation to the player. This can serve to provide a much more immersive game as players would not be able to fully predict their opponent's next move.

1.1.1 Motivation

AI is being applied in every sector. However, even though it is also being applied in video games, the removal of the default difficulty settings could allow the possibility to revolutionise the gaming industry and allow games in the future to maintain the player's focus throughout by having not just a game to play, but a game which adjusts the experience to the player's actions in realtime. This means that the modern games of today could adjust the world presented to the player, according to their performance, from the enemy's skills to the environment around them.

NPCs would respond to the player's actions in a unique and unpredictable way with the progress of machine learning. This would ultimately reduce the need for hard-coded logic in video games saving time for the developers to refine other areas of the game. Recently, something similar has been implemented in Metal Gear Solid V (2015) which tracks the player's style of gameplay and counters it. For example, if the player is tactical and is constantly using a sniper from afar, the NPCs generated would use helmets to counter it. However, this only affects the weapons used, and the enemies still followed hard-coded logic therefore leaving them to be predictable.

There is also Hello Neighbour (2017) which uses AI to track the player's tactics and counter them in the next run. The AI first tracks the player's past actions and then sets traps along the path to block the path from being exploited in the next attempt. Although this was a major step for AI in video games, this still was not done in realtime.

1.2 Hypothesis and Research Questions

The main hypothesis of this research study was that if machine learning agents were applied to replicate enemy behaviour in a video game, a player could have an optimised experience while playing and be engaged more with the game. From this, the following research questions were proposed:

- Can machine learning agents be trained as adversaries?
- Can the machine learning agents provide a balanced experience which is neither too easy nor too challenging?
- Can the machine learning agents provide a believable experience?
- Can the game provide an engaging experience to the player?

The first research question will be tackled by training agents to play against each other during the development of the game. This will be done using the ML-Agents toolkit and if successful, then it will be answered accordingly.

The rest of the research questions will be answered by the participants from the survey which will be given to them after they play the prototype developed.

1.2.1 Organisation of Study

1.2.1.1 Literature Review

The Literature Review (found in Chapter 2) in this study will go over the following topics:

- Challenge in Video Games
- Dynamic Difficulty Adjustment
- Machine Learning
- Training Unity ML-Agents

This review gives an outline of the current practices and new technologies which can be implemented to make the prototype possible. It also discusses certain ideologies and techniques which may help improve the player experience.

1.2.1.2 Methodology

The methodology (found in Chapter 3), will explain how the prototype will be developed, and assessed and ultimately how the results will be obtained and interpreted.

1.2.1.3 Results

The results (found in Chapter 4) will analyse the results obtained from the participants and be categorised by a number of factors to see if the player experience was enhanced for the general audience, not just a specific target audience.

1.2.1.4 Conclusion

The conclusion (found in Chapter 5) will identify any potential for further investigation and study, as well as assess whether the research questions and hypothesis have been addressed appropriately and if they were correct.

Chapter 2

Literature Review

2.1 Video Games

In recent years, video games have been ever increasing in popularity; this can be seen from the very beginning with the start of the first Pong game in 1972. Thus, video games have become a vital part of our society especially in today's day and age. More specifically, games tend to have a feature where the main aim is for the player to achieve a certain goal to win. Therefore, games offer a set of challenges which players need to overcome, level by level. Such challenges usually provide various difficulty standards which cater to everyone's playing ability. Games like Contra and Super Mario Bros.: tend to cater to everyone's difficulty requirement with ties to the era that they came out. Most times The Lost Levels are seemingly unwinnable and do not offer a range of difficulty choice. Modern games, on the other hand, like Kingdom Hearts 3 and Shadow of the Tomb Raider make it possible for the players to choose certain different settings to accustom to their difficulty level. This dissertation covers and analysis how video game difficulty has altered throughout the years and how such changes has evolved to make it a more enjoyable experience for players, regardless of their knowledge of experience in video games.

2.1.1 Impact of Challenge in Video Games on Players

Although it may be tedious, difficulty in video games has been proven to be beneficial for a lot of people. From better hand-eye coordination to problem-solving skills, it helps players approach day-to-day tasks in a more logical manner and ultimately further enhances their lives. It was also deduced that playing challenging games which offer a positive failure strategy had increased players' persistence and effort through failure. This resulted in achieving better grades in school. People who play video games are often better equipped at persisting when failing a challenge. [2] Challenges in video games have also shown that players felt more satisfied after finishing the game. Games have also shown that while facing functional challenges, players often overcome emotional challenges. [3] In another study conducted to monitor the impact of video games on players' behaviours, it was found that challenge and a high skill level requirement within a video game, resulted in the player being more engaged and immersed within the game itself. [4] Therefore it has been proven in multiple studies, that challenge and engagement in video games have a positive impact on players both from a mental perspective as well as an emotional perspective.

2.1.2 Difficulty Adjustment in Video Games

In the past it was not always the case that players had options to choose from to change their difficulty settings, as is nowadays seen. Games like Pacman, Metroid and Zelda provided no options of different difficulty levels. Games released today have multiple different settings such as easy, medium and hard settings as well as more complex levels for experienced veteran players, as well as easier levels such as "Story Mode" which allows players to enjoy a more relaxing experience throughout the entirety of the game if they choose. This can be witnessed in God of War Ragnarok (2022). To further explain, the game gives the players a ”Give Me Story” difficulty option which lowers the defence and damage stats of the enemy and even allows bosses to be stunned easily and reduces the frequency of attacks as well. However the difficulty option ”Give me God of War” is also present. This gives bosses a stat advantage with having higher defence, damage and stuns immunity, while also having the frequency of attacks of both bosses and enemies be continuous and multiple opponents can also attack simultaneously. [5] Another game which has done this is Kingdom Hearts 3 ReMind. At the start of the game, it allows players to pick and choose from a list of ”EZ Codes” which make the game significantly easier to ”PRO Codes” which go as far as to disable power-ups and even disallows players to heal themselves. [6] Therefore games are already allowing players to have the option of customising the difficulty within their game experience for them to have the most pleasurable and engaging experience possible.

2.1.3 Opponents in Video Games

Opponents in video games are usually always present. Be it a simple enemy shooting at you from behind a cover to a complex boss countering the player’s moves and forcing them to use meta tactics they may not be fully used to, all these require the opponent to be controlled by a Game “AI”. The term AI is not always used when complex neural networks of anything of that sort are included. One of the most common techniques is the use of control opponents in games which is the application of a Finite State Machine (FSM). An FSM divides an opponent’s behaviour in logical states which perceive the object to have natural and intelligent behaviour. The logical states usually vary according to information gathered from the player’s actions, for example when the enemy sees the player or when the player attacks. However information about the enemy itself can also be useful in switching states, for example when the enemy health is low and it needs to heal or run away from the battle. This technique was used throughout popular video games such as Age of Empires, Quake and Doom among others.

[7]

2.2 Challenge in Video Games

Video games offer a variety of challenges. Depending on the type of game, a platformer requires focus and concentration to jump between one platform and another, a first person shooter requires tactical thought to avoid getting hit by the opponent, even a simple clicker game usually hides items and require creative thinking and assessment of clues available at hand to gather what you need to proceed to the next stage. Challenge is present in all games regardless of the genre. Therefore such an aspect requires to be taken into consideration when developing a game for players to have a balanced experience whilst playing.

2.2.1 Difficulty in relation to User Experience

When presenting a challenge in a video game, the difficulty has to be balanced with the player's abilities. When reaching this balanced state, previous researchers have referred to this as the flow zone. The flow zone allows the player to be fully immersed within a video game to the point where they would feel a loss of self consciousness as well as losing sense of how much time has passed while playing. [1]

As can be seen in the first graph of Figure 2.1, if the challenge is greater than the player's abilities, it would result in the player feeling anxious. However if the challenge is too low and the player's abilities surpass the current scenario, then the player might feel bored. Therefore the balance between the two would avoid the player feeling either one of these and instead be playing in the moment without any negative feelings. [1]

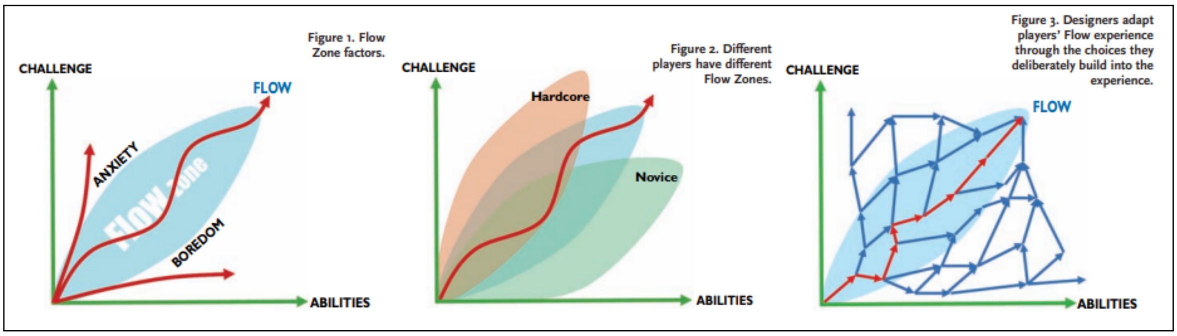


Figure 2.1: Flow Zone Graphs [1]

With each different player, the flow zone tends to differ accordingly. One player may have an averages skill set and therefore handle the challenge at hand well enough. A novice gamer on the other hand may find it too difficult while a hardcore gamer will tend to find it easy. Difficulty needs to be adjusted and catered according to the players skills while playing the game. This can be seen in the second graph of Figure 2.1. [1]

To achieve this, the researcher has proposed to integrate game mechanics to force the player unknowingly to make choices of whether they make it easier or more challenging for themselves. This would create a more interactive experience while at the same time not constantly pestering the player with how they are currently finding their game experience. In the third graph of Figure 2.1, it shows that this would constantly alter the challenge to fit within the flow zone. This would result in a seamless difficulty adjustment which would happen in the background without the player even realising. [1]

2.3 Dynamic Difficulty Adjustment

A technique which can help keep the player within the flow zone is known as DDA. It has been proposed and implemented in video games in recent years. DDA is a technique for real-time automatic adjustments to scenarios, parameters, and player behaviour in video games, which prevents player boredom or frustration when the game is too easy or too hard. DDA aims to keep the player's interest and provide a satisfying challenge level throughout the game. [8]

To achieve Dynamic Difficulty, the player's skills have to be assessed. For this to be done, firstly some variables need to be gathered and monitored. These variables usually correlate with the game rules and the win-or-lose conditions. For example in Pac-Man, the number of times the player hits the walls, direction switches, key presses, lives lost and pills collected within a specific time can indicate the player's current skill set and an assessment can be made. The assessment provides a reference point that helps the game adjust itself accordingly. Throughout the gameplay, data analysis must be carried out on the variables gathered and a level of 'ease' is found. [9]

The technique of dynamic difficulty is ultimately an alternative to preset channels of difficulty widely used in a great number of video games. When applied to a multiplayer online battle arena (MOBA), dynamic difficulty proved to be quite accurate. With a success rate of 85%, where the agent managed to keep up with the player's skill. To evaluate how well-matched the agent was, the game was played by a variety of people and then the players completed a survey about their experience. The survey involved questions about the challenge within the game, to assess the agent's capability, as well as whether the player would play the game again and if they would recommend it to others. This helps show how effective the application of dynamic difficulty was on the player's experience. [10]

2.3.1 Benefits and Drawbacks of Dynamic Difficulty

Dynamic difficulty as a technique provides numerous benefits to the gaming community both for video game developers as well as video game enthusiasts.

When adjusting difficulty, this is usually done to change the player's emotional state. Be it too bored or too frustrated, maintaining the flow zone is the ideal position to keep the player fully entertained. As shown in the previously mentioned study, players who had their agent successfully match their skills, were more likely to enjoy the game and recommend it to others. [1] [10]

Therefore if the game leaves that much of an impact on players, and the fact that players are more likely to recommend the game would benefit greatly the developers as well.

However that being said, this technique does carry some heavy drawbacks with it. Due to the massive number of computations and being a heavy drain on the system resources, it may not be suitable to adapt each game to work with dynamic difficulty especially considering the massive load most AAA video games already have on the system. [11]

Therefore dynamic difficulty, for now, should best be kept for games which have low system requirements.

2.3.2 Previous Applications of Dynamic Difficulty

To produce dynamic difficulty within a game, previous projects must be looked into to identify their solution, their logic and ultimately, how all this can be improved upon.

The agent made to add dynamic difficulty to a MOBA title was produced by first adding means to track the player's performance. This was tracked by deducting the player's death count from the player's level and then adding the number of towers destroyed. Monitoring this value throughout the duration of a match could give an outline of how the player's performance is progressing. Players picked to play against the agent, varied in age and education however all had a background in video games and had played MOBA games at least once. To adjust the difficulty accordingly, three different levels of agents were produced and the evolution of the player's performance was tracked. If the performance was being kept at the same pace, then it means that the user is not being engaged properly. [10]

In another project, a tower defence game was produced and dynamic difficulty was implemented within it which alters the enemies' stats such as health. By using the player's lives, skill points and the enemies' health, the difficulty is adjusted and each wave becomes more accustomed to the player's current play style. [12]

2.4 Machine Learning

Machine learning is one of the many applications or type of artificial intelligence which is the use of algorithms to parse data, subsequently learn from it, and make predictions or decisions which centre around the data given or provided. This is a type of adaptive technology which adheres to a specific set of features in order for the system to improve performance over time without being programmed.

Such machine learning algorithms are trained and evolved on large datasets which provide the opportunity to identify patterns to make decisions or predictions. The aim is for these algorithms to be based on a flexible and adaptable manner in order to improve their performance over a period of time without the need of ongoing programming.

This technique has already been applied in a large variety of sectors from computing tools such as image recognition and speech recognition to even agricultural, medical diagnoses and music. Current learning systems involve learning by being programmed, which is basically written to perform a particular task, learning by memorisation where the algorithm is given data which it processes into information by itself, learning from examples which information is provided and fed into the algorithm directly by the user with labels for the algorithm to process it. There is also an unsupervised method of learning which allows the algorithm to observe and discover by itself using discovery systems and theory formation tasks. [13]

2.4.1 Machine Learning in Video Games

Machine learning can be applied in different manners when it comes to video games. A common use is the generation of non-player characters known as NPCs. These are used to provide allies or otherwise opponents for the player. Machine learning algorithms can be trained on data acquired from human behaviour and decision-making to generate NPCs which behave more realistically to mimic and counteract player's actions.

Machine learning can also be used in video games to make predictions of player behaviour and provide a more dynamic game difficulty. This provides a more personalized and engaging experience for the players and such data allows for better adjustments according to player performance and skill level for a more seamless experience.

Unity has also added to this by developing the ML Agents framework. Through this tool, developers have managed to provide a similar experience to players of all skill levels, promote skill differentiation in terms of score, and do so using minimal resources relative to the deterministic AI. This makes it a more effective and efficient tool for creating engaging and personalized gaming experiences. [14]

2.4.2 Unity ML Agents

Unity ML-Agents is a framework for training machine learning models in the Unity game engine. Such a systems makes it possible to create intelligent agents which are trained to take actions based on decisions in a virtual environment.

Unity ML-Agents framework use a type of reinforcement learning, which is a type of machine learning that involves agents learning through trial and error in order to provide maximum reward signal. This makes it possible for agents to adapt and learn through its environment to achieve goals.

The framework provides tools and libraries which make it easier for developers to train and create intelligent agents in the Unity game engine. This makes it possible to create custom environments, rewards and actions for the ML-Agents, which make it possible to evaluate the performance and train the agents accordingly.

This can be applied to the field of dynamic difficulty adjustment in video games, allowing developers to create agents that can learn and adapt to the player's skill level in order to provide a more personalized and engaging gaming experience. [15]

2.4.3 Different Uses of Unity ML Agents

Unity ML Agents can therefore be used to create intelligent agents with the ability to learn and evolve through their environment with the goals to achieve rewards. When taking video games into consideration, this can be used to train and adapt to the player's skill sets to provide a more engaging and personalised experience. Player satisfaction and enjoyment can be improved greatly as well as game engagement to make it more accessible to all player's skill levels.

The use of a ML Agents was investigated in a virtual environment to control a car. Inputs, including readings from depth sensors, the car's current speed and position, and its relative position to the target, were fed into the neural network. Outputs were interpreted as engine force, braking force, and turning force. The AI began with random behavior and gradually learned to solve the task by responding to environment feedback in the form of rewards and penalties. A reward was given for getting closer to the parking spot and reaching it, while penalties were issued for driving away from the parking spot or crashing into obstacles. The final reward for reaching the parking spot depended on the car's orientation relative to the actual parking position. [16]

The ML-Agents Toolkit provide example environments where agents play soccer and tennis. The agents use what is called adversarial training. Self-play uses the agent's current and past 'selves' as opponents, providing a naturally improving adversary for the agent to gradually improve against using traditional reinforcement learning algorithms. The fully trained agent can be used as competition for advanced human players. [17]

2.4.4 Adversarial Training

Adversarial training is used in training machine learning models which involves the use of a second model, called the "adversary", to generate synthetic or made up data which is designed to fool the first model. Both models are trained simultaneously, with the purpose of improving accuracy of the initial model and the ability for the adversary to generate convincing synthetic data.

Such an approach has been used in various applications, including AI and video games. A game developer as an example may use adversarial training to improve the capability for an AI opponent to challenge human players in a game of chess. This involves a training model based on chess as well as another model to generate possible move which are based and designed to confuse the first model. This subsequently is done so that AI opponents are able to learn from synthetic data to become more engaging to the real opponent or player.

One potential limitation of adversarial training is the computational cost. Generating synthetic data is time-consuming and training two models at the same time, requires a lot of computational resources or power. Recent research however has shown that certain variations of adversarial training is more efficient and provides good results. An example is Fast Gradient Sign Method (FGSM). It allows for a wider adaptation of adversarial training in machine learning applications.

2.5 Training Unity ML Agents

To train Unity ML Agents, an environment has to be setup. After this is done, the agent's default behaviour, such as movement and an action, need to be programmed. Then observations are added such as distance from goal and direction facing. Once the observations are set, then a reward system is setup. [18]

The rewards are added according to each task tackled, having unbalanced points can cause the agent to prioritise actions which make them seem unbelievable or just bad opponents overall. Therefore multiple runs may be needed in order to find the ideal reward system for the agents being trained. [18]

Then to actually train the agents, an Anaconda environment needs to be setup and the mlagents package needs to be added to the virtual environment. Pytorch also needs to be added to the machine to start the training. A YAML configuration file then needs to be created with the proper training details such as batch size, epochs, layers and many other details. After this is done, then training can properly begin by running the command:

```
mlagents-learn config.yaml --run-id=run1
```

In that command, ”–run-id” refers to the name of the current run and the ”config.yaml” refers to the created config file. Training can be viewed and observed through the TensorBoard dashboard. [19]

Once training is completed, the model with the best results is saved along with a list of timings it took to train. The model can then be added to the Unity game object itself and it will run automatically. [19]

2.5.1 Adversarial Training in ML Agents

Research aimed towards developing agents who would act as opponents for the player was proposed in 2022 where the primary focus was to produce agents to play tennis. These agents were trained using adversarial training.

In this work, a training environment for intelligent agents to play tennis was suggested. The environment was made using the Unity Real-Time Development Platform, which made it possible to accurately simulate the physical interactions and logical behaviour of actual items. During the training, the environment made it easier to gather precise and diverse observations, which were then supplied to the training site using the ML-Agents Toolkit.

This package was used to perform training with reinforcement learning. The investigation led to the creation of two models. The tennis environment demanded complicated interactions from the ML agent, such as the capacity to strike a ball from any position inside a court area, which needed total flexibility of movement and rotation along all three dimensions. Furthermore, the agents also had to be able to distinguish between various types of shots from the opponents and react accordingly. Due to the large search area and weak reward signal, the second model was trained with these unlocked attributes in a different environment.

Ultimately, the training of an intelligent tennis agent resulted in an agent which behaved satisfactorily and served as a test application's non-playable opponent. [20]

2.5.2 Believability of ML Agents

To produce believable agents, research was performed attempting to employ reinforcement learning to build AI agents that can adapt to dynamically changing physics-based surroundings. Using incentive functions, the objective was to create plausible entities who displayed intrinsic behaviours, such as guarding their goal and attacking the ball. The reward function was altered via trial and error to gradually create behavioural patterns that enhance performance. It had been demonstrated through testing that agents performed better than those with less specified reward functions and state spaces. It had been discovered that the final agent developed through the tests was convincing and challenging to tell apart from a human player.

The believability of the agent was found by surveying participants with 25 questions. The questions could have been answered in three ways, they could either answer correctly or incorrectly or remain unsure. The survey was divided into two sections, the first 14 questions required participants to view gameplay footage and identify whether the character was player controlled or an agent. The rest of the questions asked the participants to evaluate the footage of an agent playing against a human in a one-on-one match and guess who was the human. [21]

2.5.3 Adaptability of ML Agents

A machine learning agent's adaptability is its capacity to function successfully in ever-changing or dynamic contexts. An agent must have the capacity to learn from its experiences and modify its behaviour as a result in order to be considered adaptable. This is crucial in many real-world applications since an agent's operating environment may alter over time.

Machine learning agents can be made more flexible in several different ways. Reinforcement learning is one of those methods, whereby an agent can learn to act or react to maximise reward signals. The agent can learn to adjust to new circumstances by changing the reward function and the environment in which it functions. Utilizing unsupervised learning algorithms is another strategy that can train an agent to identify patterns in data and predict the outcome of brand-new, unobserved data.

The design and training of machine learning agents generally take flexibility into account. An agent can become more useful in a range of situations and jobs by being more adaptive.

Adaptability was also shown to increase the immersion of a video game since the agents produced allowed the player to experience actions which were classified as unpredictable and realistic. [22]

2.5.4 Emotions of Player when playing against ML Agents

Players may feel various emotions while competing in video games against machine learning (ML) agents, including excitement, challenge, frustration, and satisfaction. Players may feel challenged and excited by the unexpected nature and high degree of skill displayed by ML agents. However, gamers could feel frustrated and let down if the agents are poorly developed. When the player is successful in taking down the agent, they are satisfied, which makes the game more enjoyable. The usage of ML agents may significantly impact the user experience in video games. When building ML agents for video games, it is essential to take the player's emotions into account in order to produce a satisfying and positive experience.

The study's findings showed that adaptive mechanisms had a positive impact on learning effectiveness and efficiency. Additionally, a variation in play behaviour might be seen. When confronted with competing agents, the participants reported a decreased sense of shame, enhanced empathy, and behavioural involvement. Calculations, however, showed no discernible effect on the mental pressures caused by potentially difficult social opponents. These findings demonstrate the potential for adaptive game systems in the future and guide the best way to incorporate social competition in various educational video games. [23]

2.6 Conclusion

Challenging video games may enhance a player's performance by providing a more satisfying and fun experience while at the same time keeping them engaged. To develop a challenging video game, introducing ML-Agents built and trained using Unity's AI Kit can be used to provide opponents which adapt to the player's skill set and learns how to provide a challenge and keep the players constantly alert. Tuning the settings of the adaptive agents may be beneficial to make the player stay within the flow zone for maximum engagement without being too difficult to too easy. The agents will then be trained using adversarial training which will focus on their adaptability, believability and their overall difficulty.

Chapter 3

Research Methodology

3.1 Introduction

With the use of Machine Learning, models can be trained to act as enemies within a game environment which would observe their surroundings to gather data for the agent to decide its actions. The agent will then be rewarded if the correct action is performed and penalised if not. Using Unity ML-Agents, game developers can train models using a pre-built framework that uses a hybrid reinforcement and imitation learning approach. Curriculum training will also be adopted in the training so that agents can learn their actions step by step instead of being expected to know their tasks immediately. This research aims to create a game which utilises these machine-learning algorithms to train believable enemy agents which use sensors to detect opponents to attack.

This research used Unity Version 2021.3.23f1. This version of Unity was chosen as it is a Long Term Support (LTS) build and is therefore reliable and secure to work with. The ML-Agents build used was the Verified Package 2.0.1 released on November 8th, 2021.

3.2 Research Methodology

The research will be conducted using a quantitative approach. The participants will first be required to play the prototype. After playing the game the participants will then be required to take part in a survey. The survey will gather some basic information about the participant such as age and gender. Such questions will help to see if the user experience is enhanced for a wide audience, not just a specific target audience. The participant is then asked whether they are proficient in games and if they are well-versed in the genre of Top Down Shooters. These questions will help establish the familiarity that participants have with games and their skill level. The second section will ask the user about the prototype itself. The user will be asked about the difficulty of the game, the believability of the agents and the engagement of the game. After all the questions were answered, the participants will be asked to optionally fill in any general feedback they have for the prototype.

3.3 Heuristic Evaluation

To figure out the heuristics of the prototype, the finished game was tested out by 8 people in pre-testing. These participants included 4 males and 4 females from different age groups with mixed experiences in gaming. Whilst playing the game, they were observed and their feedback was noted. Some feedback was also obtained regarding the difficulty manager's thresholds. The bugs encountered were fixed before the next participant tried it out next. This was the feedback obtained from the players:

Type	Description	Participant	Status
Bug	Controls issue in movement where player orientation affected movement	1	Fixed
Suggestion	Number of minimum enemies should be increased as it was boring	2	Fixed
Suggestion	Max damage for enemies should have a threshold as they were overpowering me easily	3	Fixed
Suggestion	The minimum accuracy threshold should be lowered	3	Fixed
Bug	Spikes not spawned on any difficulty	3	Fixed
Bug	Bomb trap did not turn off on the easiest difficulty	4	Fixed
Suggestion	Faster enemy spawns on higher difficulties	5	Not implemented
Bug	Explosion indicators caused damage to the player	6	Fixed
Suggestion	Increase bullet damage for player	7	Fixed
Suggestion	Sound effects	8	Not implemented

Table 3.1: Pre-Test Feedback

3.4 Prototype

This section will provide a description of how the ML-Agents were set up, trained and implemented in the game.

3.4.1 ML-Agents

Running episodes where the Agent tries to complete the job is a part of the ML-Agents Toolkit training process. The Agent appears in each episode until it either dies (runs out of health) or times out (takes too long to solve or is unsuccessful at the mission).

To train the model, first, an agent script was created and called EnemyAgent. The functions of this script included the basic actions of the enemies, which were to rotate around, move up and down, move left and right, and fire. This script also implemented the base Agent class from the ML-Agents toolkit. Therefore the agent also implemented the functionalities for episodes, sensors and observations and actions.

This script was then attached to a simple game object with a cube mesh and a simple gun mesh made of cubes attached to it. The agent was also given a ray perception sensor 3D. The game object was also given a Behaviour Parameters component and a Decision Requester. A Prefabricated Object (Prefab) was created from the game object then and put in an arena against a player prefab with the same script.

3.4.1.1 Behaviour Parameters

The enemy behaviour was called ”EnemyAgent”. It was given 0 space size for vector observations but instead the ”Use Child Sensors” was ticked since all necessary observations were obtained from the sensor itself. The agent was given 3 continuous actions, to look around, move horizontally and move vertically, and a single discrete branch to fire with a size of 2, since the options are to either fire or not fire. To view how these were used, refer to Appendix B.1.

3.4.1.2 Ray Perception Sensor

The sensor was given a list of detectable tags. These included the player tag, the enemy tag, a wall tag and tags for enemy and player projectiles alike. 25 rays per direction were set to be emitted at all times with a max ray angle of 180 degrees. The ray length was set to 60 so the entirety of the arena would be covered and the rest of the settings were kept the same. The sensor set-up can be seen in Appendix A.

3.4.1.3 Agent Class

In the script, the methods implemented from the interface included the OnEpisodeBegin and OnActionRecieved methods.

The OnEpisodeBegin method refilled the health of the enemy and picked a random spawn point from a list of spawners set within the arena. The episode ended when the max step set in the behaviour parameters was reached or when the agent had its health reduced to 0.

The OnActionRecieved method used the actions received by the decision requester to calculate the next action. The decision requester had a **decision period** set to 5, which allowed the agent to choose its action every 5 steps. To view the decision requester, refer to Appendix B.3. The actions were then passed to the method. In order to adopt curriculum learning, the agents had multiple boolean checkers to check whether they should fire, they should rotate, they should move horizontally and they should move vertically. The continuous actions had values of floats between -1 and 1. This meant that for example if the action currently being decided is to look either left or right, a value of -1 would make the agent look left while a value of 1 would make the agent look right. The continuous actions array was declared using index 0 as the rotation action, index 1 as the move horizontally and index 2 as the move vertically. To view how these were used, refer to Appendix B.2

3.4.1.3.1 Curriculum Learning

Curriculum Learning is a strategy used to train models in Machine Learning. Tasks of increasing difficulty are given to the model to train on similar to the way humans and animals learn. They first start off with similar concepts and then begin getting more complex tasks to tackle. This technique is used as studies have found that it can significantly increase what is learnt by the model by allowing it to first understand easy aspects of a task and then slowly progress to start tackling more difficult ones. [24]

3.4.1.4 Rewards System

The agents were trained using Curriculum Training. These were the parameters set for each training stage:

Training Stage	No. of Enemies	No. of Players	Should Look	Should Fire	Move Z	Move X	Walls	Health	Random Stats
1	1	1	Yes	No	No	No	No	Yes	No
2	1	1	Yes	Yes	No	No	No	Yes	No
3	1	1	Yes	Yes	Yes	No	No	Yes	No
4	1	1	Yes	Yes	Yes	Yes	No	Yes	No
5	1	1	Yes	Yes	Yes	Yes	Yes	Yes	No
6	2	1	Yes	Yes	Yes	Yes	Yes	Yes	No
7	4	1	Yes	Yes	Yes	Yes	Yes	Yes	No
8	4	1	Yes	Yes	Yes	Yes	Yes	Yes	Yes
9	4	1	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Table 3.2: Training Stages and their characteristics

For training session 1, the agents were rewarded for looking at each other. They were given a max step of 200 so that the agents would not waste time looking elsewhere.

For training session 2, the agents were rewarded when they fired at their opponent. They were penalised for missing and when hit, the agent was respawned at a random position once its health was depleted or when the max step was reached. This time a max step of 300 was set. For the following sessions whenever the adversary's health was depleted, the agent was rewarded an extra bonus.

For training session 3, the agents were allowed to move on the Z-Axis, they were rewarded for hitting their adversary, however, they were penalised when getting hit. The max step was set to 1000 here allowing the agents to move freely and consider their actions further. The agents were also penalised when touching a wall in order for them not to constantly stay within the borders of the map.

For training session 4, the agents were allowed to move freely within the arena and they were given a max step of 2000 so that time was taken to constantly look at the target and fire. The agents were rewarded based on hitting the player with the reward increasing by distance and were penalised for missing. The agents were also rewarded for covering some distance over time to entice them to roam around instead of staying in a single area.

For training session 5, walls were enabled within the arena. The max step was increased to 3000 since the agents now had to learn to look for their opponents as well.

For training session 6, once the agents were used to moving about, shooting and finding their opponent, the agents were trained with 2 enemies. For the time being, the agents were still being trained to attack the player only and were therefore penalised when attacking each other. The rewards system remained the same. The player was also given more health so that the training was realistic.

For training session 7, the number of enemies was increased to 4 and were still focused on attacking the player. The rewards system was kept the same as the previous session.

For training session 8, in order to simulate the dynamic difficulty, random stats were given to the agents on each episode. The rewards were kept the same.

For training session 9, the arena setting was embellished and they were rewarded for attacking each other just the same. The agents were equipped with a method to find all adversaries within the arena and set their adversary according to who was closest to them. This allowed the agents to have a more realistic approach in not only attacking the player but each other as well.

3.4.2 Difficulty Manager

The difficulty manager was vital to make the game dynamically difficult. This removed the usual difficulty settings set in normal games and instead altered each stat according to the player's performance.

3.4.2.1 Enemy Stat Changes

The stats of the enemies were dynamically altered based on the player's performance. These stats included the enemy's minimum and maximum damage, the enemy's health, the enemy's movement speed and the health pickups' drop rate.

Changes were done based on the player's accuracy, the number of enemies defeated and the player's health. For example, if the player's accuracy was high, then the enemies were buffed and had increased health and damage, while if the accuracy was low, then the enemies were weakened and their health and damage were reduced.

3.4.2.2 Environmental Changes

3.4.2.2.1 Spike Traps

The difficulty was also adjusted using the environment around the player. The traps which were manually placed around the arena were set to disabled at the beginning of the game. These traps involved a mesh of cubes placed within each other to include a spike-like shape.The traps were randomly divided into three levels called TrapsLvl1, TrapsLvl2 and TrapsLvl3 alike. These levels were activated according to the level of difficulty currently triggered by the difficulty manager.

3.4.2.2.2 Bomb Spawner

In order to entice the player to move, a bomb spawner trap was added to the game. The bomb spawner emitted a ray at a random direction from the player, and if it found that the area was empty, a bomb was spawned. The bomb was a simple prefab which first spawned a cube which changed colour from white to red repeatedly, then the cube shrunk while still changing colours to indicate that the bomb was close to detonation. Once that period was over, the bomb disappeared and a yellow sphere spawned and expanded slowly to imitate an explosion. The distance from the player to spawn the bombs at and the time to detonate could be adjusted according to the difficulty.

3.4.2.2.3 Light Manager

A light manager script was set up and attached to a game object. The player was given a torch and a background light. This helped make the environment more mysterious and dangerous as the visibility was very limited. Two methods were implemented to make this possible, a MakeDarker method followed by a MakeLighter method. Using Unity's render settings' ambient intensity, the light of the entire arena was either reduced to make it darker or increased to make it lighter. These methods were triggered according to the player's accuracy.

3.4.2.3 Stat Tracking

Tracking the player's stats was an essential part of this script's process. The stats were used to measure the player's performance and adjust the game's difficulty accordingly. The stats kept note of were the player's accuracy, the player's health and the number of enemies killed.

3.4.2.4 Overall Works

The script started off by randomly dividing the spike traps into 3 separate lists. From there a default beginning stats were set. These stats were determined by some pre-evaluation testers of the game. The move speed was set to 10, the drop rate was set to 100, a minimum damage of 2 and a maximum damage of 5 were set, the health was set to 10 which meant that the enemy took 2 bullets to be defeated. The difficulty was set to medium and the number of enemies concurrently spawned was set to 4.

Three coroutines were then started, one which tracked the player's stats, one which checked the number of enemies currently in the arena and one which checked the current difficulty. A **coroutine** is a function used within Unity which opens a seperate low memory thread which allow for non-sequential execution. These can be paused, resumed and delayed for a specific value of time.

3.4.2.4.1 TrackStats Coroutine

The track stats coroutine ran once every 5 seconds. When run, the accuracy was first calculated by checking the number of shots hit. If the shots hit was 0, the accuracy would count to be invalid, so that the accuracy would not be divided by 0. Then the number of shots hit would be divided by the number of shots fired and an approximate value would be obtained. Then the current health would be obtained and the number of enemies killed. This coroutine also contained a variable which tracks how much the difficulty was either increased or decreased.

If the player accuracy was greater than 0.8, the enemies' health and damage were increased and the light was decreased. The difficulty count would also increase by 1. On the other hand if it was less than 0.3, than the difficulty count would be decreased by 1 and the enemies' health and damage were decreased as well and the light was increased. If the accuracy was set to -1 (no shots were hit) these conditionals would not be triggered.

Then the enemies killed were checked. If the enemies killed were greater than 3, than the number of enemies was increased along with the difficulty count. However if the player did not kill any enemies and the number of shots hit was not equal to 0, then the number of enemies would be decreased and the enemies were made weaker along with the difficulty count being reduced by 1. The enemies were made to be weaker by reducing their health and damage.

The player's health was then checked, and if it was less than 20% then the enemy damage was decreased and the enemies' drop rate for health pickups was increased. If on the other hand, the health was greater than 80%, then the enemy's damage was increased and the drop rate for the health pickups was decreased.

After all this was done, the counts for the accuracy and enemies killed were reset and the function was delayed for 5 seconds before running again.

3.4.2.4.2 CheckEnemies Coroutine

This coroutine finds all present enemies within the arena and stores a total number. If the number of enemies is less than the number of enemies which are supposed to be present, then a SpawnEnemy coroutine is called. This works by spawning an enemy in a random spawn point from a list of enemy spawn points. The coroutine also set all the enemy stats as the current stats to ensure that the newly spawned enemies have the same stats as the rest of the currently spawned enemies. The enemies are spawned at an interval of 3 seconds if more enemies need to be spawned.

3.4.2.4.3 CheckDifficulty Coroutine

The CheckDifficulty Coroutine runs once every 15 seconds, it checks the difficulty count accumulated from the previously run TrackStats coroutine and increases the game difficulty if the count is positive or decreases the difficulty if the count is negative. The difficulty would then be triggered in the game. The difficulties were set up as follows:

- Easy
- EasyMedium
- Medium
- MediumHard
- Hard
- Harder
- Hardest

These are how the difficulties were set up:

Difficulty Stage	Traps Level 1	Traps Level 2	Traps Level 3	Bomb Spawner	Trap Damage	Health Pickup
Easy	Off	Off	Off	Off	0	10f
EasyMedium	Off	Off	Off	Spawns every 5 seconds	5	8%
Medium	On	Off	Off	Spawns every 4 seconds	7	7%
MediumHard	On	Off	Off	Spawns every 3 seconds	9	6%
Hard	On	On	Off	Spawns every 2 seconds	10	5%
Harder	On	On	Off	Spawns every 1 second	12	4%
Hardest	On	On	On	Spawns every 1/2 second	15	3%

Table 3.3: Difficulty Stages and their characteristics

Chapter 4

Analysis of Results and Discussion

4.1 Introduction

Quantitative findings of the player's experience with the prototype will be presented and discussed in this section. The feedback was obtained using a survey which was completed after playing the game. The questions are presented in the form of statements to the participants which gives the participants the option to fully agree and fully disagree using the Likert Scale. The survey discussed can be found in Appendix C.

4.1.1 Likert Scale

The Likert Scale was deemed to be the best option for this study as it allows the participants to express their opinions clearly and the data gathered would be uniform. This would also prove to help when the statistics were analysed. The Likert Scale also helps in the validity of data as it captures a value of intensity on their opinions not just agree or disagree. This way the data gathered will also be more reliable as it allows for a consistent measurement when compared to a simple yes or no answer. The scale also provides an easy-to-use grade system, instead of having the participants give a finite grade of their opinion. [25]

4.2 General Information Questions

These questions can be found in Appendix C.1 and Appendix C.2. The first 2 questions gather data about the participant's gender and age which are useful as it is in this study's best interest to enhance user experience for a wide audience.

In this study, 71.1% of the participants were male, 28.9% of the participants were female and 2.6% did not disclose their gender. For the age, 7.9% were between 13-17 years old, 42.1% were 18-24 years old, 36.8% were 25-34 years old, 10.5% were 45-54 years old, and 5.3% were 55-64 years old.

As for the game experience, 63.2% considered themselves to be proficient video game players, 26.3% did not, and 10.5% considered themselves to be neither proficient nor not experienced at all. As to the specific genre of the game, from the participants, 44.7% of themselves thought themselves to be well-versed in the genre of top-down shooter video games, while 52.6% did not. 2.6% considered themselves to neither be proficient nor not experienced.

4.3 Main Questions

These questions provided a rating for the difficulty of the game in relation to the participant's skill level, the enemies' believability and reactions and the engagement level of the participant towards the game. These questions can be found in Appendix C.3.

4.3.1 Difficulty Rating

The difficulty rating was obtained by asking the participants if they agreed with the following statement ”The game presented an appropriate level of difficulty tailored to your abilities.”

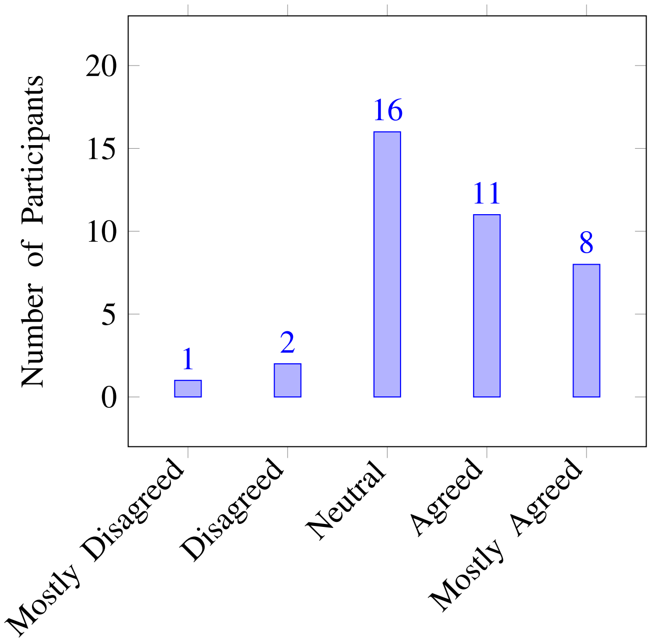


Figure 4.1: General Difficulty Rating

The majority (42.1%) voted that they neither agreed nor disagreed with this statement. While 28.9% agreed with this statement and 21.1% mostly agreed. 5.3% disagreed with this statement and 2.6% mostly disagreed. Overall, the game provided a difficulty rating average of 3.61.

This showed that although the game presented an appropriate level of difficulty for a lot of the participants, this could still be improved upon as there was quite a large number who did not find it adequate for them.

4.3.2 Adversary Rating

The difficulty rating was obtained by asking the participants if they agreed with the following statement ”The game presented an appropriate level of difficulty tailored to your abilities.”

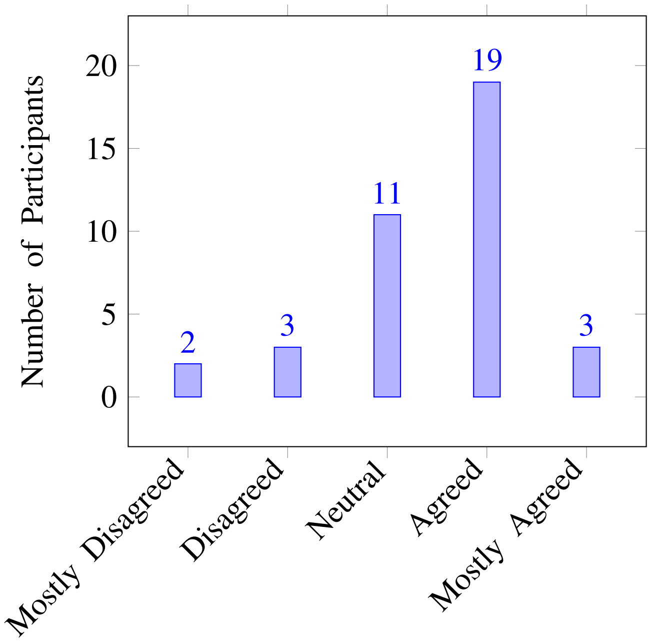


Figure 4.2: Adversary Rating

The majority (50%) voted that they agreed with this statement. While 28.9% were neutral with this statement. 7.9% mostly agreed and 7.9% disagreed with this statement. 5.3% mostly disagreed with this statement. Overall, the agents achieved an adversary rating average of 3.5.

This showed that the agents were mostly perceived to be believable by the participants. However, they were still not given a full rating by the participants and this shows that the agents could still have been improved upon. A large number was also neutral towards this statement and some even disagreed showing that the agents’ actions were not believable enough for all players.

4.3.3 Engagement Rating

The engagement rating was obtained by asking the participants if they agreed with the following statement ”The gaming experience was engaging, and successfully maintained your focus throughout the entire game.”

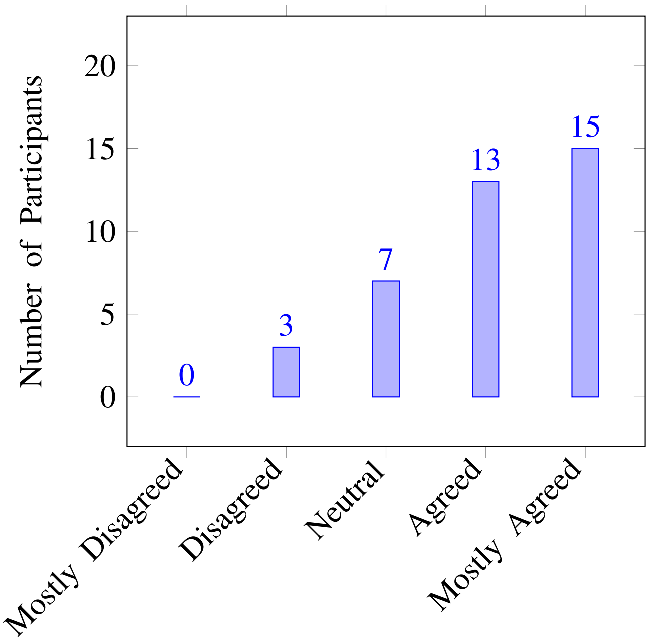


Figure 4.3: Engagement Rating

The majority (39.5%) voted that they mostly agreed with this statement. While 34.2% agreed with this statement and 18.4% were neutral. The rest (7.9%) disagreed with this statement and no one mostly disagreed with this. Overall, the game provided an engagement rating average of 4.

This showed that the game was engaging for the large majority of players. A small number disagreed with this statement and some even felt neutral towards this statement showing that although the techniques used did engage the player, it still could have been improved.

4.4 In-Depth Analysis of the Main Questions

The following sections will discuss the ratings by gender, age group, game experience and genre experience. However, the categories do not all have the same number of participants (e.g. gender has 27 males, 11 females and 1 undisclosed) these figures will be represented as percentages in the analysis and discussions and the average results will be taken into consideration.

4.4.1 Ratings by Gender

These are the average ratings categorised by gender:

Gender	Average Difficulty Rating	Average Adversary Rating	Average Engagement Rating	Count
Male	3.6	3.5	4	26
Female	3.7	3.5	4.4	11
Other	3	3	3	1

Table 4.1: Summary of Results by Gender

4.4.1.1 Difficulty Rating by Gender

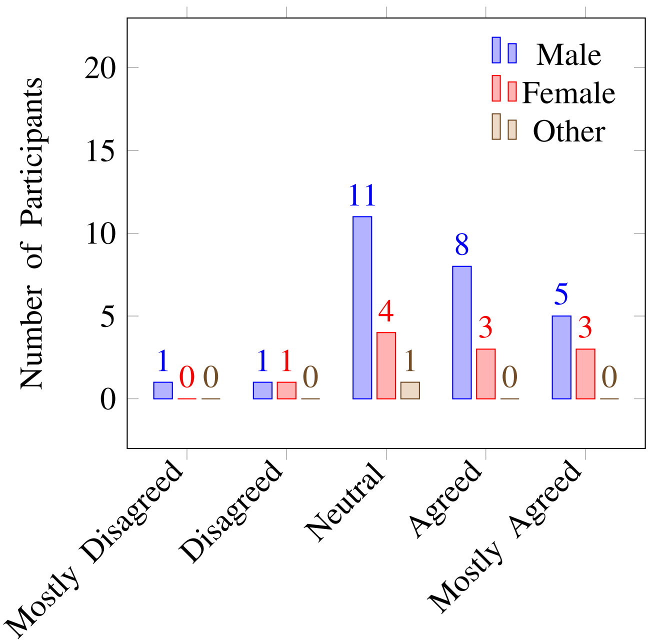


Figure 4.4: Difficulty Rating by Gender

From what can be observed when categorising the difficulty ratings by gender, 3.84% of male participants mostly disagreed, 3.84% of male participants and 9.1% of female participants disagreed. 42.3% of male participants and 36.4% of female participants were neutral which showed that they felt that they did not have an adequate level of difficulty presented to them but was close. 30.8% of male participants and 27.3% agreed that the game presented an adequate level of difficulty based on their skills. Finally, 19.2% of male participants and 27.3% of female participants found it perfectly balanced for them. Overall, male participants gave a difficulty rating average of 3.6 while female participants gave an average of 3.72.

This showed that female participants found the game to be slightly more adequately difficult for them when compared to males since the average obtained by the female participants was 3.7 while the male participants' average was 3.6. For the participant who did not disclose their gender, their opinion was neutral and got an average of 3 showing that the difficulty was not as accustomed to them.

4.4.1.2 Adversary Rating by Gender

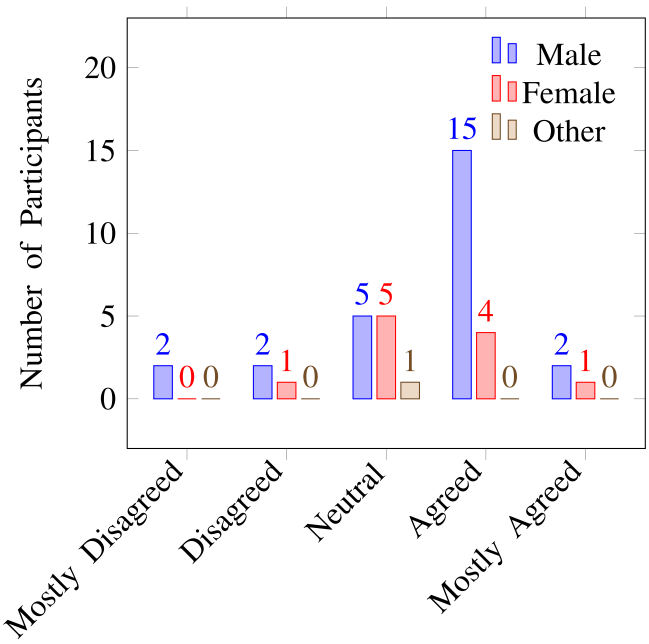


Figure 4.5: Adversary Rating by Gender

From what can be observed when categorising the adversary ratings by gender, 7.7% of the male participants found the agents to be not believable at all. Another 7.7% of male participants and 9.1% of females found the agents somewhat not believable too. 19.2% of the male participants and 45.5% of the female participants found the agents to be neither believable nor not believable as did the single participant who did not disclose their gender. 57.7% of the male participants and 36.4% of the female participants found the agents to be somewhat believable. 7.7% of the male participants and 9.1% of the female participants found them to be mostly believable. Overall, male participants gave the adversaries an average rating of 3.5 while female participants gave the adversaries an average rating of 3.45.

This shows that overall, both male and female participants found the agents to have the same level of believability with both genders grading the agents with a believability rating of 3.5.

4.4.1.3 Engagement Rating by Gender

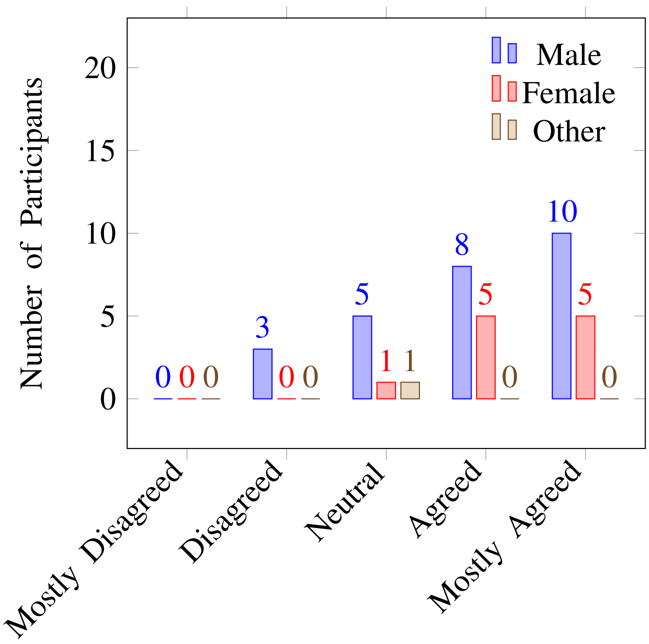


Figure 4.6: Engagement Rating by Gender

From what can be observed when categorising the engagement ratings by gender, 11.5% of the male participants found the game to be somewhat not engaging. Another 19.2% of male participants and 9.1% of females found the game to be engaging but not so much, as did the single participant who did not disclose their gender. 30.8% of the male participants and 45.5% of the female participants found the game to be quite engaging. 38.5% of the male participants and 45.5% of the female participants found the game to be fully engaging. Male participants gave the game an engagement rating of 3.96 while the female participants gave the game an engagement rating of 4.36. This shows that the average engagement rating between the male and female participants was quite similar but when comparing them together, female participants found themselves to be more engaged in the game than male participants.

Overall, although the agents were not as believable and the difficulty may not have been adequate, the game still managed to provide an engaging experience regardless of gender.

4.4.2 Ratings by Age Groups

Note that from the age groups presented, no participants were of under 13 years of age, 35-44 years old or 65 years and older. The age groups that will be discussed are 13-17 years old, 18-24 years old, 25-34 years old, 45-54 years and 55-64 years old. These were the average ratings categorised by age groups:

Age Group	Average Difficulty Rating	Average Adversary Rating	Average Engagement Rating	Count
13-17 Years Old	3.3	3	4.3	3
18-24 Years Old	3.8	3.5	4.2	15
25-34 Years Old	3.4	3.5	3.8	14
45-54 Years Old	3.8	3.5	4.5	4
55-64 Years Old	3.5	4	3.5	2

Table 4.2: Summary of Results by Age Groups

4.4.2.1 Difficulty Rating by Age Groups

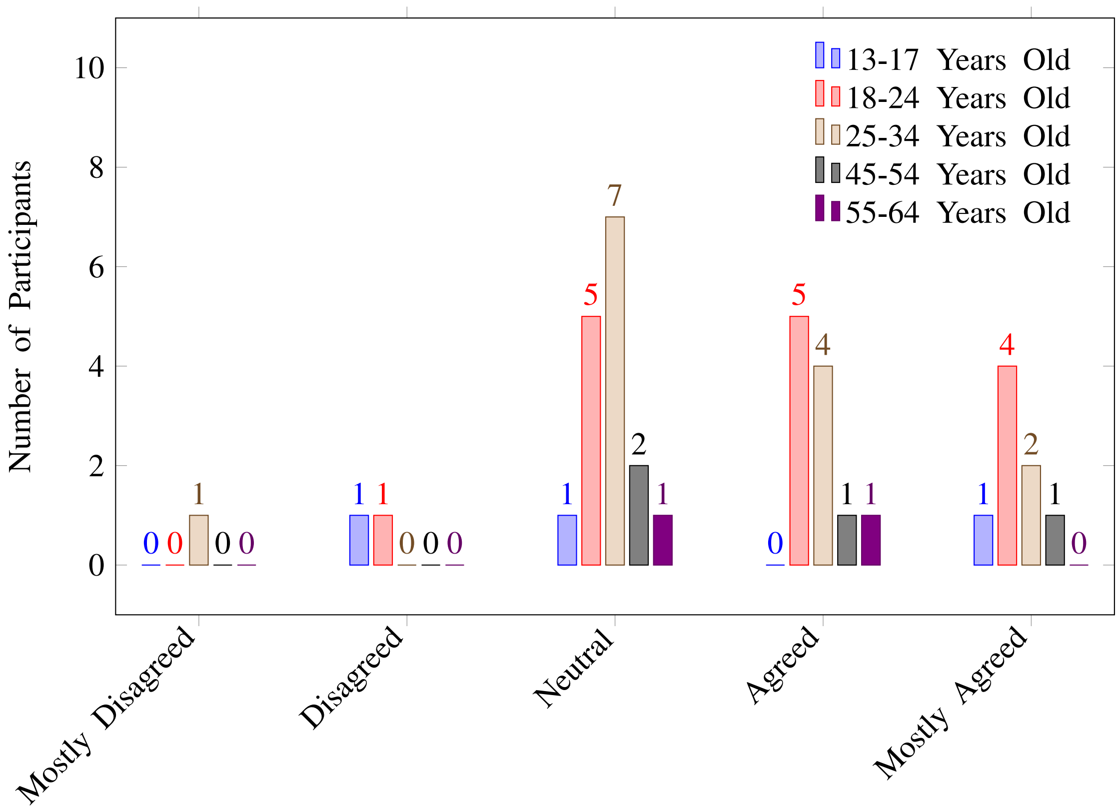


Figure 4.7: Difficulty Rating by Age Groups

From what can be observed when categorising the difficulty ratings by age groups, it seems that 7.1% of 25-34 year old participants found the game to not match their skill level at all. 33.3% of 13-17 year-old participants and 6.6% of 18-24 year-old participants found the game inadequate in matching their skill level. 33.3% of 13-17 year old participants, 33.3% of 18-24 year old, 50% of the 25-34 year old participants, 50% of the 45-54 year old participants and 50% of the 55-64 year old participants found the game to be neither adequately difficult nor inadequately difficult. 33.3% of the 18-24 year old participants, 28.6% of the 25-34 year old participants, 25% of the 45-54 year old participants and 50% of the 55-64 year old participants, found the game to be quite adequately difficult for them. 33.3% of the 13-17 year old participants, 26.7% of the 18-24 year old participants, 14.3% of the 25-34 year old participants and 25% of the 45-54 year old participants found the game to be perfectly adequately difficult for them.

This shows that with age, the difficulty stayed more or less the same across all age groups with the minimum average being a 3.3 from the 13-17 year old participants and the maximum average being a 3.8 from the 18-24 year old participants and the 45-54 year old participants.

4.4.2.2 Adversary Rating by Age Groups

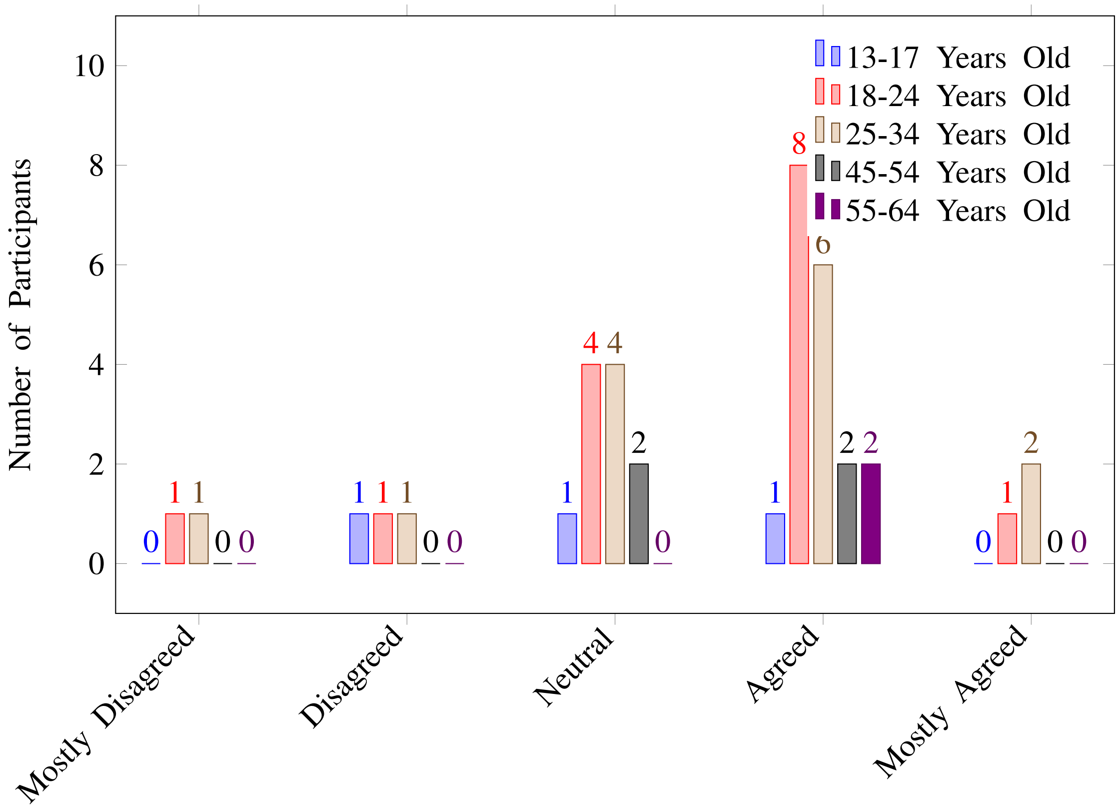


Figure 4.8: Adversary Rating by Age Groups

From what can be observed when categorising the adversary ratings by gender, 7.7% of the male participants found the agents to be not believable at all. Another 7.7% of male participants and 9.1% of females found the agents somewhat not believable too. 19.2% of the male participants and 45.5% of the female participants found the agents to be neither believable nor not believable as did the single participant who did not disclose their gender. 57.7% of the male participants and 36.4% of the female participants found the agents to be somewhat believable. 7.7% of the male participants and 9.1% of the female participants found them to be mostly believable.

This shows that the females somewhat had higher expectations from the believability of the agents when compared to the male participants.

4.4.2.3 Engagement Rating by Age Groups

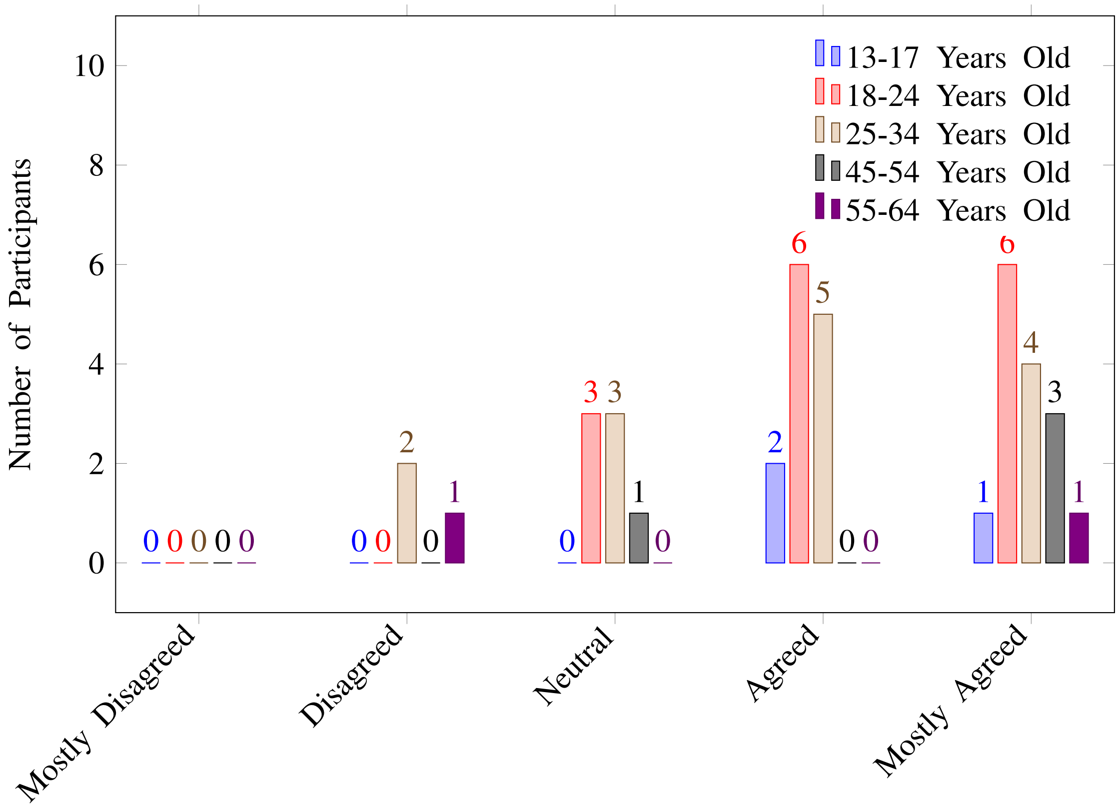


Figure 4.9: Engagement Rating by Age Groups

From what can be observed when categorising the engagement rating by age groups, 14.2% of the 25-34 year old participants and 50% of the 55-64 year old participants found the game to not be as engaging. 20% of the 18-24 year old participants, 21.4% of the 25-34 year old participants and 25% of the 45-54 year old participants found the game to not be adequately engaging. 66.6% of the 13-17 year old participants, 40% of the 18-24 year old participants and 35.7% of the 25-34 year old participants found the game to be quite engaging.

33.3% of the 13-17 year old participants, 40% of the 18-24 year old participants, 28.6% of the 25-34 year old participants, 75% of the 45-64 year old participants and 50% of the 55-64 year old participants found the game to be fully engaging for them.

This shows that engagement was the least maximised with the 55-64 year old participants and was fully maximised with the 45-54 year old participants. This however does not show much difference in the ranges in average with the minimum being 3.5 and the maximum being 4.5.

Overall, age was not really a huge factor when it came to the difficulty, adversary and engagement as the averages were quite similar for all the groups.

4.4.3 Ratings by Gaming Experience

These are the average ratings categorised by gaming experience:

Gender	Average Difficulty Rating	Average Adversary Rating	Average Engagement Rating	Count
Yes	3.6	3.5	4	24
No	3.6	3.6	4	10
Do not know	3.5	3.25	4.75	4

Table 4.3: Summary of Results by Gaming Experience

4.4.3.1 Difficulty Rating by Gaming Experience

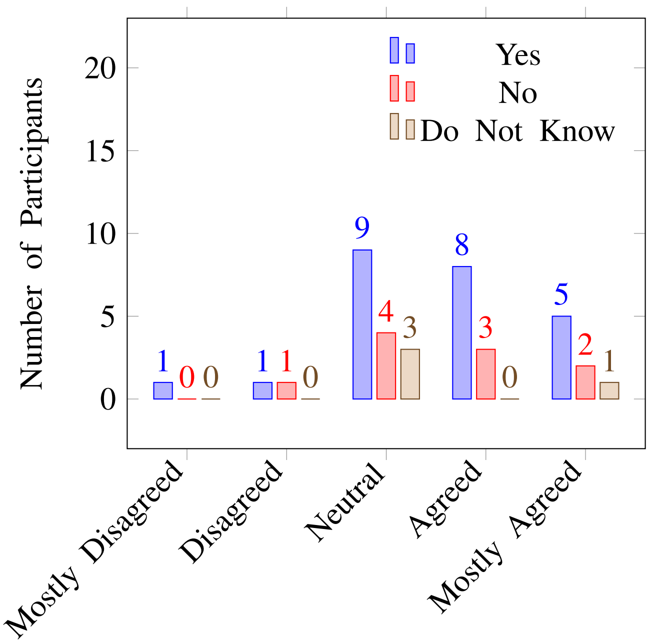


Figure 4.10: Difficulty Rating by Gaming Experience

From what can be observed when categorising the difficulty ratings by gaming experience, 4.2% of experienced gamers participants found that the difficulty was not adequate for them. Another 4.2% of experienced gamers participants and 10% of non-experienced participants disagreed that the difficulty was not so adequate either. 37.5% of experienced gamers participants, 40% of non-experienced gamer participants, and 75% of the unsure participants were neutral which showed that they felt that they did not have an adequate level of difficulty presented to them but was close. 33.3% of the experienced gamers participants and 30% of the non-experienced gamers agreed that the game presented an adequate level of difficulty based on their skills. Finally, 20.8% of experienced gamers participants, 20% of non-experienced participants and 25% of unsure participants found it perfectly balanced for them.

Overall, both experienced gamers participants and non-gamers participants gave a difficulty rating average of 3.6 while those unsure gave an average of 3.5.

This showed that female participants found the game to be more adequately difficult for them when compared to males. As for the participant who did not disclose their gender, their opinion was neutral and therefore got an average of 3 showing that the difficulty was not fully accustomed for them.

4.4.3.2 Adversary Rating by Gaming Experience

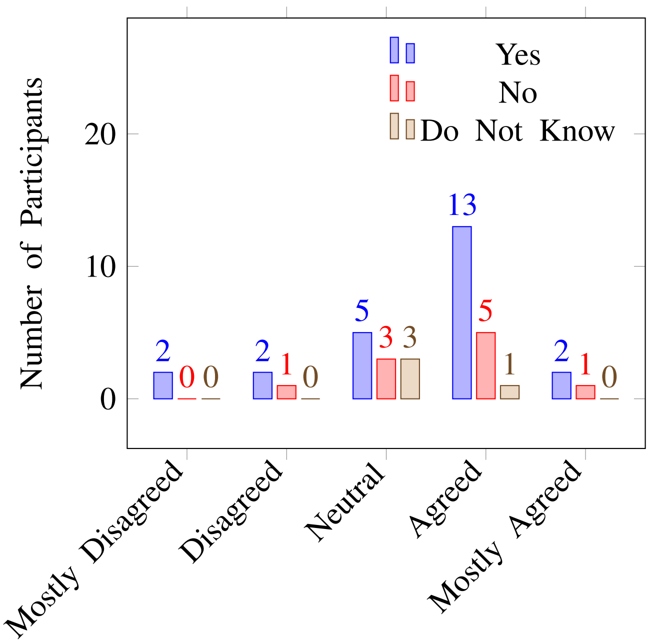


Figure 4.11: Adversary Rating by Gaming Experience

From what can be observed when categorising the adversary rating by the gaming experience, 8.3% of the experienced gamers participants found the agents' actions to not be believable at all. Another 8.3% of the experienced gamers participants and 10% of the non-experienced gamers participants found the agents to not be as believable either. 20.8% of the experienced gamers participants, 30% of the non-experienced gamers participants and 75% of the unsure participants found the agents to be somewhat believable. 54.2% of the experienced gamers participants, 50% of the non-experienced gamers participants and 25% of the unsure participants found the agents to be believable while 8.3% of the experienced gamers participants and 10% of the non-gamers participants found the agents to be fully believable. Overall, the difference between the experienced and non-experienced gamers participants was somewhat negligible being that they got an average of 3.5 and 3.6 respectively. Those unsure however found the agents less believable with an average of 3.25. This shows that there was not much correlation between the agents' believability and the participants' gaming experience.

4.4.3.3 Engagement Rating by Gaming Experience

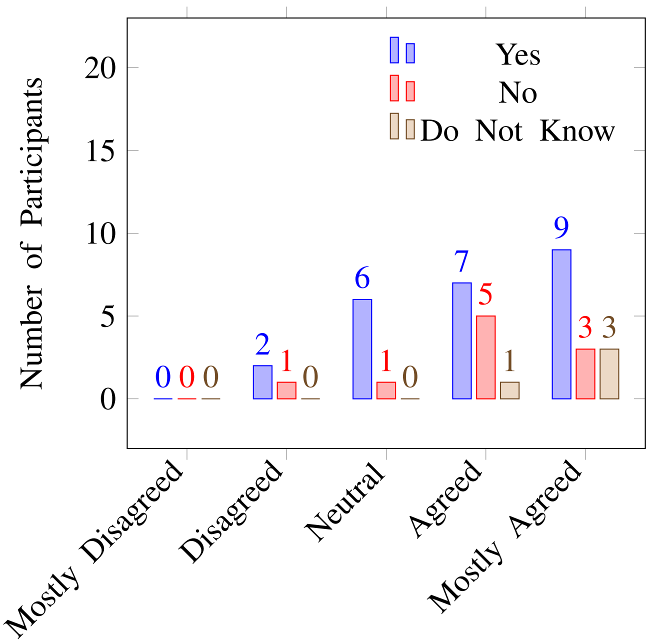


Figure 4.12: Engagement Rating by Gaming Experience

From what can be observed when categorising the engagement ratings by gaming experience, 8.3% of the experienced gamers participants and 10% of the non-experienced gamers participants found the game to not be as engaging. Another 25% of experienced gamers participants and 10% of non-experienced gamers found the game to be engaging but not so much. 29.1% of the experienced gamers participants and 50% of the non-experienced gamers participants found the game to be quite engaging as did 25% of those who were unsure of their gaming expertise. 37.5% of the experienced gamers participants, 30% of the non-experienced gamers participants and 75% of those unsure of their gaming experience found the game to be fully engaging. Both experienced and non-experienced gamers participants got an engagement rating of 4 while those unsure got an even greater rating of 4.75.

This shows that the participants were fully engaged regardless of the participants' gaming experiences.

4.4.4 Ratings by Genre Experience

It should be noted that only a single participant was unsure of their experience in the top-down shooting genre. These are the average ratings categorised by genre experience:

Genre Experience	Average Difficulty Rating	Average Adversary Rating	Average Engagement Rating	Count
Yes	3.5	3.2	3.9	17
No	3.6	3.7	4.1	20
Do Not Know	5	3	5	1

Table 4.4: Summary of Results by Genre Experience

4.4.4.1 Difficulty Rating by Genre Experience

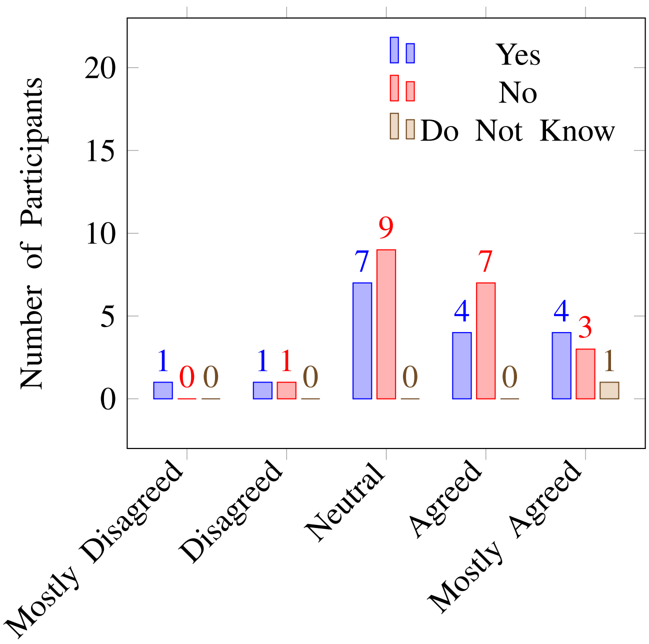


Figure 4.13: Difficulty Rating by Genre Experience

From what can be observed when categorising the difficulty ratings by the participants' experience in the genre, 5.9% of those experienced found the difficulty to not be adequate at all. Another 5.9% of those experienced and 5% of those not experienced also found the difficulty to not be as adequate. 41.1% of those experienced and 45% of those non-experienced found the difficulty to be somewhat adequate for them. 23.5% of those experienced and 35% of those not experienced found the difficulty to be adequate enough for them. Another 23.5% of experienced participants, 15% of non-experienced participants and the single unsure participant found the game to provide the right amount of difficulty for them.

Overall this shows that with the genre experience, the difficulty stayed more or less the same with only a 0.1 difference in the averages where the experienced achieved an average difficulty rating of 3.5 and the inexperienced achieved a rating of 3.6. The single unsure participant found it fully adequate and received a difficulty rating of 5.

4.4.4.2 Adversary Rating by Genre Experience

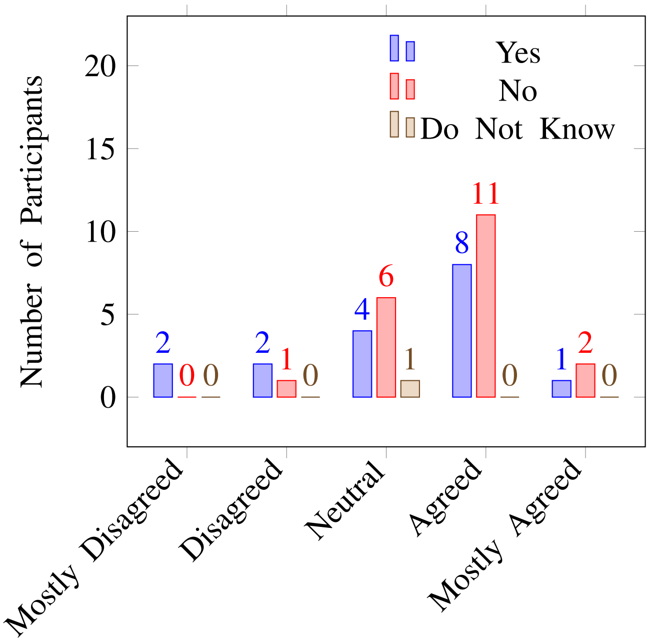


Figure 4.14: Adversary Rating by Genre Experience

From what can be observed when categorising the adversary ratings by the participants' experiences in the genre, 11.8% of the experienced participants found the agents to be not believable at all. Another 11.8% of the experienced participants and 5% of non-experienced found the agents somewhat not believable too. 23.5% of the experienced participants and 30% of the non-experienced participants found the agents to be neither believable nor not believable as did the single unsure participant. 47.1% of the experienced participants and 55% of the non-experienced participants found the agents to be somewhat believable. 5.9% of the experienced participants and 10% of the non-experienced participants found them to be mostly believable. This shows that the experienced participants somewhat had higher expectations from the believability of the agents when compared to the non-experienced participants since the averages were 3.2 for the experienced and 3.7 for the non-experienced. The single participant who was unsure got an average of 3.

4.4.4.3 Engagement Rating by Genre Experience

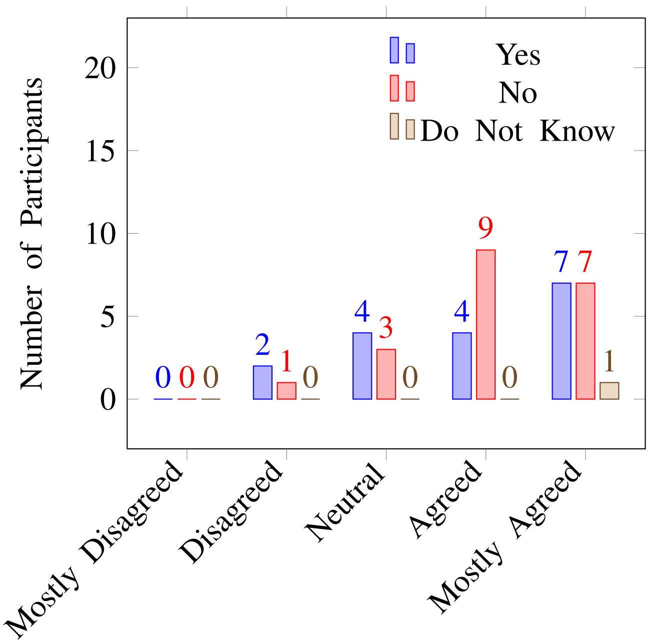


Figure 4.15: Engagement Rating by Genre Experience

From what can be observed when categorising the engagement ratings by the participant's experience in genres, 11.8% of the experienced participants and 5% of the non-experienced participants found the game to be somewhat not engaging. 23.5% of experienced participants and 15% of non-experienced participants found the game to be engaging but not so much. Another 23.5% of the experienced participants and 45% of the non-experienced participants found the game to be quite engaging. 41.2% of the experienced participants and 35% of the non-experienced participants found the game to be fully engaging as did the single participant who was unsure of their experience in the genre. This shows that the engagement was mostly similar for both those experienced and those not experienced in the genre.

4.5 Discussion

4.5.1 Can Machine Learning Agents be trained as Adversaries?

The first research question in this research is if it is possible to train ML-Agents as adversaries.

In order to answer this question, the term adversaries needs to be defined. In the context of this research, adversaries meant that the agents would use what they are learning against each other in order to improve their behaviour to ultimately be able to participate in combat against a player.

The ML-Agents were successfully trained as adversaries, first having them fight a single opponent, and then slowly escalating the environment to a free for all environment. They found their opponents and targeted them till their opponent's health was depleted or they themselves were defeated.

The agents presented to the players in the game utilised an adversarially trained neural network. The neural networks worked accordingly although there were still some issues present where a few participants stated that sometimes the AI ”did not detect” them or that they ”were randomly shooting and moving around”.

However, the overall feedback was positive and therefore it shows that the agents were successfully trained as adversaries.

4.5.2 Can the Machine Learning Agents provide a balanced experience which is neither too easy nor too challenging?

The difficulty of the game was controlled throughout gameplay using a difficulty manager.

The difficulty manager included functionalities to implement environmental changes. These were performed in the manner of spike traps, bombs spawned close to the player as well as lighting effects to reduce enemy visibility for the player. The participants found the traps to be well-placed, however, a number of participants were not sure of what the bombs were even though these were explained in the tutorial. Some participants also mistook the lighting effects as a day-night cycle but it was still appreciated.

Besides environmental changes, the difficulty manager included stat changes in the enemy agents, trap damage, bomb spawn rate and even health pickups drop rate and heal amount. These stats different seemed to be efficient and no feedback was received regarding this which therefore showed that most participants seemed to not notice this or were not affected negatively by it.

The overall difficulty rating was 3.61. This showed that although the majority seemed to have a positive experience with the difficulty, this could still be improved upon. Therefore the answer to this research question is yes, the agents were able to provide a mostly balanced experience to the participants.

4.5.3 Can the machine learning agents provide a believable experience?

Believable is a broad term. In this case believable means that the agents choose the appropriate actions in any scenario they encounter.

The participants provided an overall rating average of 3.5. Therefore, for the most part, the agents did provide a believable experience for the participants. They attacked the player and each other as was instructed similar to an arena fashion. However, as mentioned previously, there were instances where the agents shot at walls or did not attempt to follow their adversary. There were also instances where the agents wandered aimlessly without focusing on a specific target.

This shows that although the agents were believable and had realistic reactions to the scenarios presented to them, they could still have been improved upon. However, it should also be mentioned that if these agents replicate player behaviour, similar to multiplayer games, when the agents were wandering around aimlessly, this could be similar to new players who are still getting used to the game. Having agents display a variety of reactions shows a form of realism in itself.

4.5.4 Can the Game Provide an Engaging Experience to the Player?

As mentioned in the literature review, engagement is a term where players are provided with an experience between too difficult and too easy called the flow zone.

The average rating participants gave for the engagement was 4, meaning that for the most part, the game provided an overall engaging experience even with the to-be-improved dynamic difficulty and enemies' behaviour.

Some feedback provided by the participants to make it even more engaging included was to include a ”score system”, ”power-ups”, ”varying enemy types” and even an ”enemies killed counter”. These features could have been implemented were it not for time constraints or being beyond the scope of the research.

There were also some participants who experienced issues with the controls and the user interface. The way the player fired was by firing every time the mouse button was clicked. Some participants found this to be tedious and would have preferred to keep pressing the button to fire. This was done more by chance but a choice between the 2 could have been implemented to accustom to a wider audience. Other participants also encountered an issue with the crosshair placement being either below or above the actual shooting position. These issues have reduced immersion unfortunately for some participants, however, the high average shows that this was not as negatively effective.

4.6 Literature Comparison

The following values were obtained when looking at another study which researched the implementation of machine learning adversaries in video games. The difficulty level obtained from the ML Agents had an average challenge rating of 2.69, an average enjoyment rating of 2.69 and an average interest rating of 2.38. [26]

Firstly the agents developed within the study were implemented within a first-person shooter(FPS) game. Although normally FPS games may be quite difficult for beginner players to adapt to, the participants in this study consisted of mostly experienced FPS players. This could also however mean that when playing the game, participants were more highly critical of the agents having been used to other games enemies having more believable actions.

4.6.1 Difficulty Rating

When comparing the results, it can however be observed that the average difficulty rating for the current research was significantly better with a global average of 3.64 when making an average, of all the averages obtained from the categories (i.e. gender, age, game experience and genre experience). This shows that regardless of age, gender, gaming experience and genre experience, the difficulty was accustomed to the participants more than adequately. [26]

4.6.2 Believability Rating

When comparing the results, it can be observed that the average believability rating for the current research was also significantly better with a global average of 3.41 when making an average, of all the averages obtained from the categories similar to the difficulty rating above. This shows that regardless of age, gender, gaming experience and genre experience, the enemy agents were perceived to have mostly believable reactions. [26]

4.6.3 Engagement Rating

When comparing the results, it can be observed that the average believability rating for the current research was also significantly much better with a global average of 4.1 when making an average, of all the averages obtained from the categories similar to the difficulty rating and the believability rating above. This shows that regardless of age, gender, gaming experience and genre experience, the participants felt engaged whilst playing the game against the enemy agents. [26]

4.6.4 Discussion

The genres of the games developed for the cited study and the current research are completely different. Since FPS games allow you to see the environment through the player's 'eyes', participants might have had a closer look at the agents and this in itself could have allowed participants in the study to more closely observe the enemies. As previously mentioned, since the large majority of participants were also experienced gamers, they could have had a higher standard of expectations and felt bored while playing the game as the agents were subpar in their opinions.

Chapter 5

Conclusions and Recommendations

The aim of this research was to see if it was possible to improve player experience with the use of Unity ML-Agents. To prove if this is true, the following research questions were produced:

- Can machine learning agents be trained as adversaries?
- Can the machine learning agents provide a balanced experience which is neither too easy nor too challenging?
- Can the machine learning agents provide a believable experience?
- Can the game provide an engaging experience to the player?

In order to tackle these questions, a prototype was developed by the researcher. The prototype used ML-Agents in place of state-based enemies and a difficulty manager in charge to customise the experience according to the player's skills. The game placed the player in an arena and required them to survive for 5 minutes.

The prototype was tested on 38 participants of mixed backgrounds to check if the player experience is enhanced for the general populous, not just a specific target audience.

5.1 Hypothesis Results

The main hypothesis of this research study was that if machine learning agents were applied to replicate enemy behaviour in a video game, a player could have an optimised experience while playing and be engaged more with the game. After observing the results, it showed that the players were immersed quite well when playing the game.

Furthermore, some participants even left feedback in the survey saying ”Game was interesting, traps where well placed and very challenging game map” and ”It was very immersing”.

5.2 Further Improvements

5.2.1 Limitations

From other feedback gathered during the questionnaire, these were the limitations encountered:

- ”When playing on my old computer the enemies did not show up, however when I tried it on a modern pc it worked. Good work!”
- ”Its not the best calibrated so you need to aim lower or higher to reach your target”
- ”Did not work on a Mac, had to test it using a family member’s device.”

5.2.2 Improvements and Suggestions

The improvements discussed by participants and testers with all mixed backgrounds were as follows:

- Sound effects and music should have been implemented in the game. Sound effects are a common practice in video games and therefore adding sound effects could have made it a more pleasant experience for the participants and testers alike.
- The crosshair image was slightly off-centred and this resulted in the aim being slightly off. If remedied, this could have improved the overall player experience even further.
- Visuals could have been improved instead of having simple shapes like cubes and spheres, better graphics and models could have made the game feel more realistic.

5.3 Research Suggestions

From this research, it was shown that ML-Agents placed in a dynamically difficult environment managed to enhance user immersion in a video game. With that being said, further research could be done on the following:

- These techniques could be applied to different game genres.
- ML-Agents could be trained using Hierarchical Reinforcement Learning which would allow a single agent to control multiple agents thus creating a form of collaboration between the enemies.

- Immersion can also be increased by allowing agents to team up with players, having multiple states such as friendly, neutral and hostile. This would allow the player to keep enemies neutral by not attacking them thus giving a more complex viewpoint on their behaviour.

- From the previous 2 points, enemies could decide to team up and be controlled by a single agent instead which would eventually provide a form of hive mind for players to combat against.

References

[1] Jenova Chen. Flow in games (and everything else). *Communications of the ACM*, 50(4):31–34, 2007.

[2] Craig G Anderson, Kathryn Campbell, and Constance Steinkuehler. Building persistence through failure: the role of challenge in video games. In *Proceedings of the 14th International Conference on the Foundations of Digital Games*, pages 1–6, 2019.

[3] Tom Cole and Marco Gillies. Thinking and doing: Challenge, agency, and the eudaimonic experience in video games. *Games and Culture*, 16(2):187–207, 2021.

[4] Muhannad Quwaider, Abdullah Alabed, and Rehab Duwairi. The impact of video games on the players behaviors: A survey. *Procedia Computer Science*, 151:575–582, 2019.

[5] Talha Abid. God of war ragnarok: All difficulty settings explained, 2022.

[6] Stefan Djakovic. Kh3 remind premium menu guide, 2020.

[7] Penelope Sweetser and Janet Wiles. Current ai in games: A review. *Australian Journal of Intelligent Information Processing Systems*, 8(1):24–42, 2002.

[8] Mohammad Zohaib. Dynamic difficulty adjustment (dda) in computer games: A review. *Advances in Human-Computer Interaction*, 2018, 2018.

[9] Gabriel K Sepulveda, Felipe Besoain, and Nicolas A Barriga. Exploring dynamic difficulty adjustment in videogames. In *2019 IEEE CHILEAN Con-*

ference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON), pages 1–6. IEEE, 2019.

[10] Mirna Paula Silva, Victor do Nascimento Silva, and Luiz Chaimowicz. Dynamic difficulty adjustment on moba games. *Entertainment Computing*, 18:103–123, 2017.

[11] Suoju He, Junping Wang, Xiao Liu, Wan Huang, et al. Dynamic difficulty adjustment of game ai by mcts for the game pac-man. In *2010 sixth international conference on natural computation*, volume 8, pages 3918–3922. IEEE, 2010.

[12] Rhio Sutoyo, Davies Winata, Katherine Oliviani, and Dedy Martadinata Supriyadi. Dynamic difficulty adjustment in tower defence. *Procedia Computer Science*, 59:435–444, 2015.

[13] Taiwo Oladipupo Ayodele. Machine learning overview. *New Advances in Machine Learning*, 2:9–18, 2010.

[14] Hongyou Xiong. *Dynamic Difficulty Adjustment: Developing an Adaptive Game AI with Machine Learning*. PhD thesis, California State University, Northridge, 2019.

[15] Unity Technologies. Machine Learning Agents — unity.com. https://unity.com/products/machine-learning-agents. [Accessed 11-Dec-2022].

[16] Ai learns to park - deep reinforcement learning, Aug 2019.

[17] Training intelligent adversaries using self-play with ml-agents. https://blog.unity.com/technology/training-intelligent-adversaries-using-self-play-with-ml-agents.

[18] Ml-agents: Hummingbirds. https://learn.unity.com/course/ml-agents-hummingbirds?uv=2019.3.

[19] Unity-Technologies. ‘unity-technologies/ml-agents: The unity machine learning agents toolkit (ml-agents). https://github.com/Unity-Technologies/ml-agents, Dec 2021.

[20] Bakhtiyar Ospanov. Training intelligent tennis adversaries using self-play with ml agents. 2021.

[21] Abdulla R Albuainain and Christos Gatzoulis. Reinforcement learning for physics-based competitive games. In *2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT)*, pages 1–6. IEEE, 2020.

[22] David J King and Cassie Bennett. An investigation of two real time machine learning techniques that could enhance the adaptability of game ai agents. In *GAMEON'2016: 17th International Conference on Intelligent Games and Simulation*, pages 41–48. EUROSIS, 2016.

[23] Steve Nebel, Maik Beege, Sascha Schneider, and Günter Daniel Rey. Competitive agents and adaptive difficulty within educational video games. In *Frontiers in education*, volume 5, page 129. Frontiers Media SA, 2020.

[24] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM, 2009.

[25] Arnold van Alphen, Ruud Halfens, Arie Hasman, and Tjaart Imbos. Likert or rasch? nothing is more applicable than good theory. *Journal of Advanced Nursing*, 20(1):196–201, 1994.

[26] Tim Nämerforslund. Machine learning adversaries in video games: Using reinforcement learning in the unity engine to create compelling enemy characters, 2021.

Appendix A

Observation Sensors

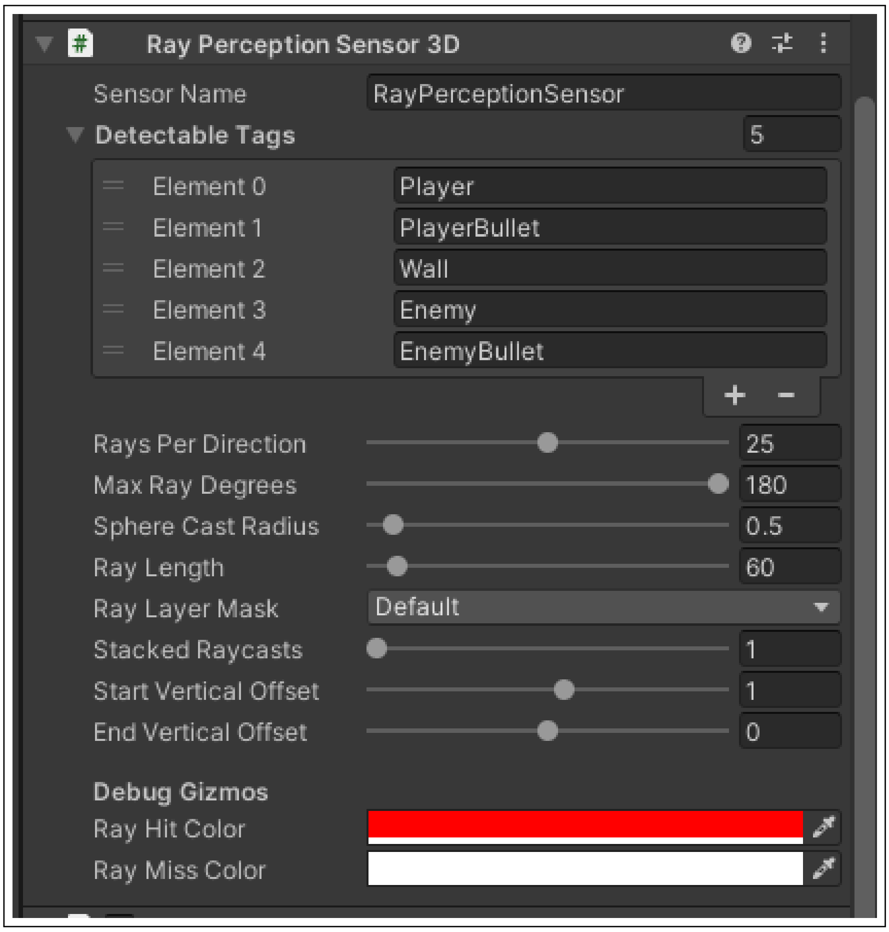


Figure A.1: Ray Perception Sensor Component

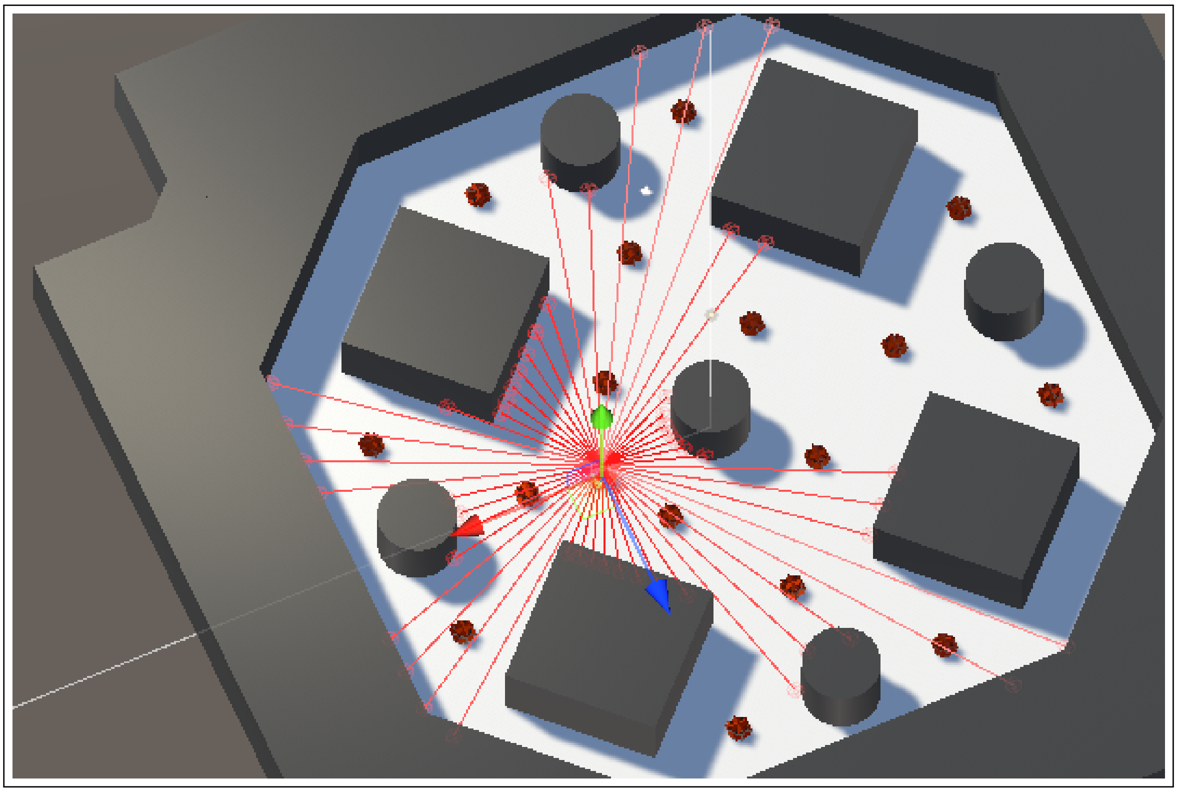


Figure A.2: Ray Perception Sensor within the environment

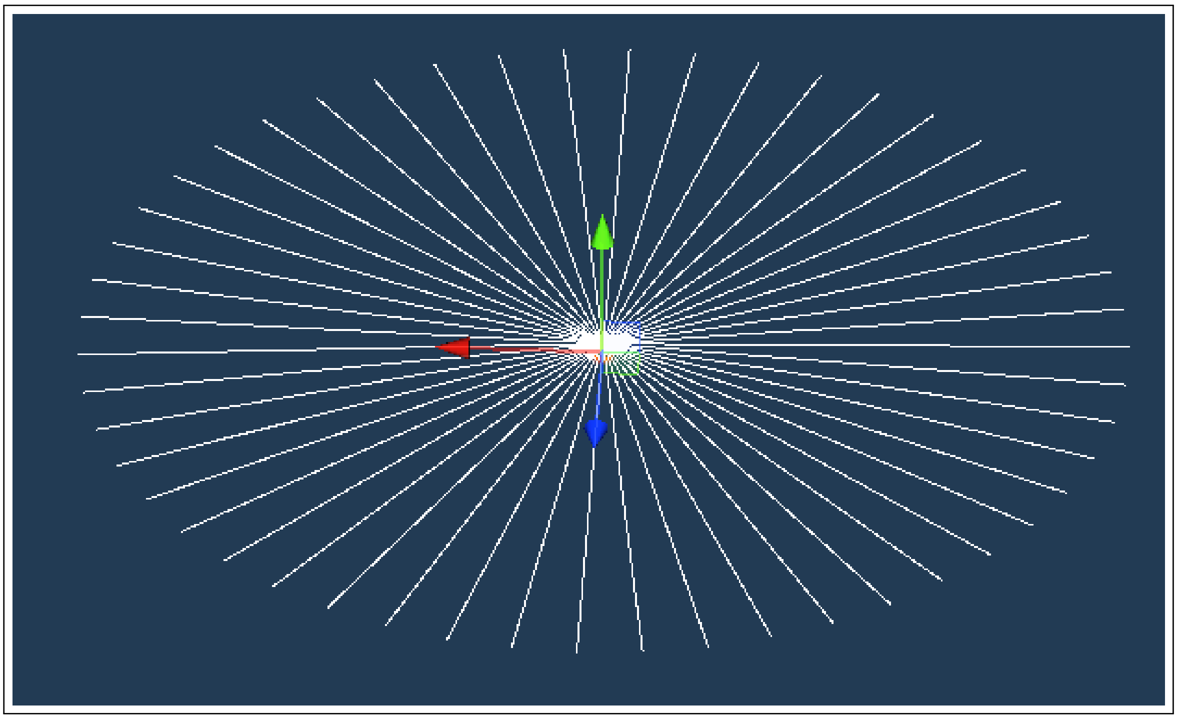


Figure A.3: Ray Perception Sensor within the Prefab

Appendix B

Behaviour Parameters

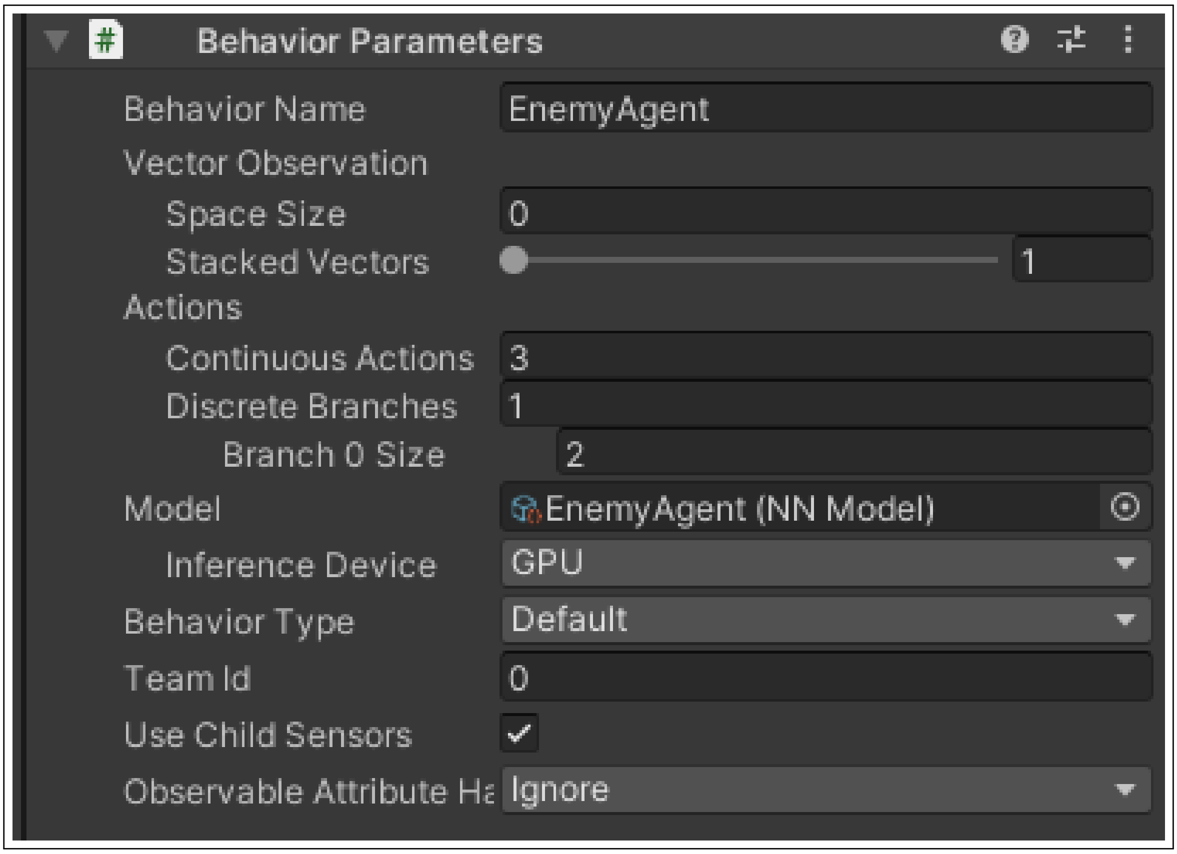


Figure B.1: Behaviour Parameters

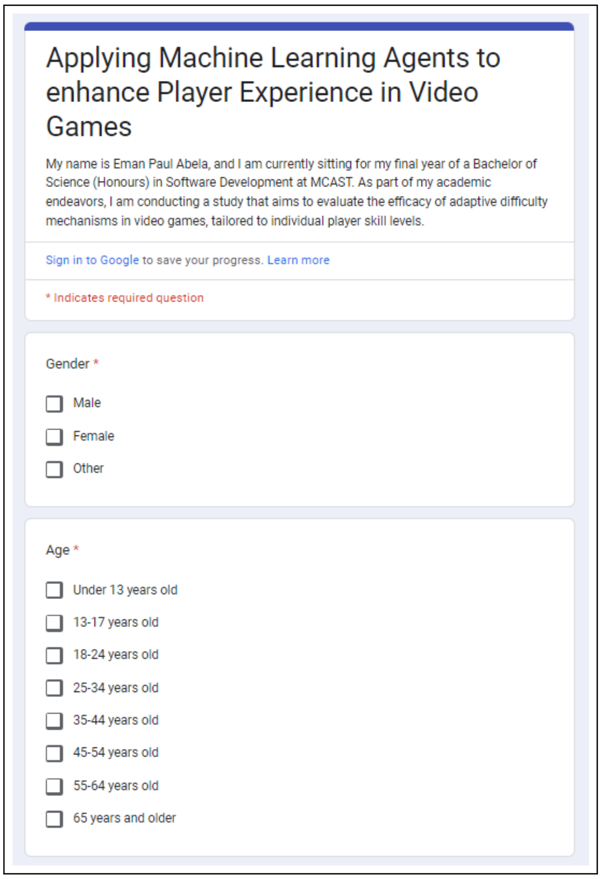
```
if (ShouldMoveX && ShouldMoveZ)
{
    HandleMovement(actions.ContinuousActions[1], actions.ContinuousActions[2]);
}
else if (ShouldMoveX)
{
    HandleMovement(actions.ContinuousActions[1], 0);

}
else if (ShouldMoveZ)
{
    HandleMovement(0, actions.ContinuousActions[2]);
}
HandleRotation(actions.ContinuousActions[0]);
```

Figure B.2: Continuous Actions

Figure B.3: Decision Requester

Appendix C

Survey



Applying Machine Learning Agents to enhance Player Experience in Video Games

My name is Eman Paul Abela, and I am currently sitting for my final year of a Bachelor of Science (Honours) in Software Development at MCAST. As part of my academic endeavors, I am conducting a study that aims to evaluate the efficacy of adaptive difficulty mechanisms in video games, tailored to individual player skill levels.

Sign in to Google to save your progress. Learn more

* Indicates required question

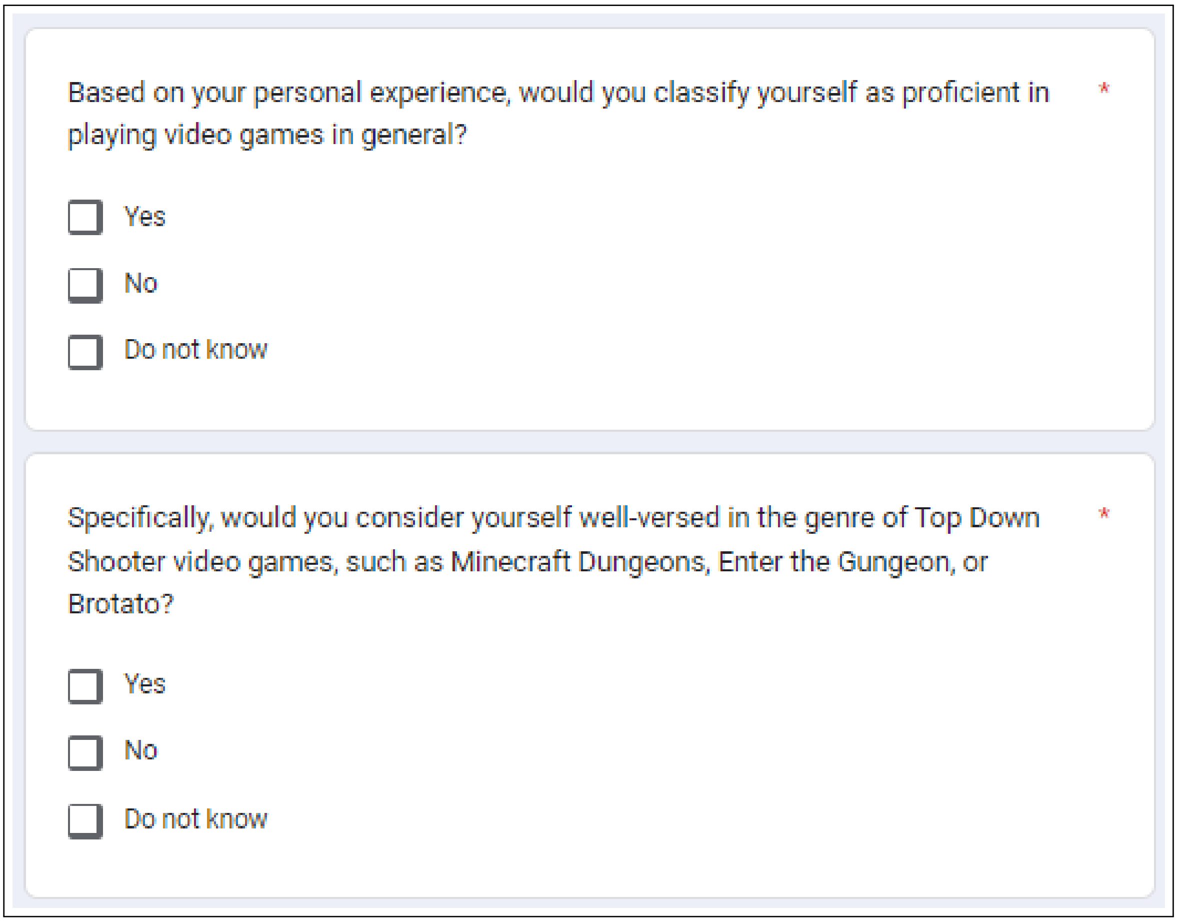
Gender *

- [ ] Male
- [ ] Female
- [ ] Other

Age *

- [ ] Under 13 years old
- [ ] 13-17 years old
- [ ] 18-24 years old
- [ ] 25-34 years old
- [ ] 35-44 years old
- [ ] 45-54 years old
- [ ] 55-64 years old
- [ ] 65 years and older

Figure C.1: Gender and Age Questions



Based on your personal experience, would you classify yourself as proficient in playing video games in general? *

- [ ] Yes
- [ ] No
- [ ] Do not know

Specifically, would you consider yourself well-versed in the genre of Top Down Shooter video games, such as Minecraft Dungeons, Enter the Gungeon, or Brotato? *

- [ ] Yes
- [ ] No
- [ ] Do not know

Figure C.2: Gaming Experience and Genre Experience

Video Game Analysis

Kindly respond to the following statements after having played the game:

The game presented an appropriate level of difficulty tailored to your abilities. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

The in-game adversaries exhibited believable behavior and reactions.

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

The gaming experience was engaging, and successfully maintained your focus throughout the entire game. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

Lastly, we would appreciate any additional thoughts or feedback you may have regarding your experience with the game.

Your answer

Figure C.3: Prototype Questions