University of Portsmouth

School of Creative Technologies

The effectiveness of learning artificial intelligence in video games.

By

Kieran Mark Grist

814853

Supervisor: Jahangir Uddin

CT6PRO

Project type: Combined

# Abstract

The project looks at and assess the effectiveness of Neurological Networks within video games, looking at the current video games and technologies. The project creates and assess a video game created with a learning intelligence inside it and determines how effective it was.

Table of Contents

[Abstract 1](#_Toc29389866)

[Terminology 2](#_Toc29389867)

[Chapter 1: Introduction 2](#_Toc29389868)

[Chapter 2: Literature Review 3](#_Toc29389869)

[2.1 Introduction into Artificial Intelligence in video games 3](#_Toc29389870)

[2.2 Introduction to Neural Networks in video games 3](#_Toc29389871)

[2.3 Neural Networks current capabilities within video games 4](#_Toc29389872)

[2.4 Main Differences between AI and Neural Networks 4](#_Toc29389873)

[2.5 Current Issues with Artificial intelligence in modern games 5](#_Toc29389874)

[2.6 Industry Problems with Neural networks 5](#_Toc29389875)

[Chapter 3: Methodology 6](#_Toc29389876)

[3.1 Time Management 6](#_Toc29389877)

[3.2 Analysing the artefact 6](#_Toc29389878)

[3.3 Software 6](#_Toc29389879)

[Chapter 4: Creating the Learning AI 7](#_Toc29389880)

[4.1 SERE 7](#_Toc29389881)

[4.2 Designing a game for an AI that learns 7](#_Toc29389882)

[4.3 Building the AI 7](#_Toc29389883)

[4.4 Training the AI 7](#_Toc29389884)

[4.5 Testing the AI 7](#_Toc29389885)

[4.6 Initial analysis of the effectiveness 7](#_Toc29389886)

[4.7 Performance 7](#_Toc29389887)

[Chapter 5: Group Study 7](#_Toc29389888)

[5.1 Selecting Members for the group 7](#_Toc29389889)

[5.2 Blind Test 7](#_Toc29389890)

[5.3 AIs adaption to the players 7](#_Toc29389891)

[Chapter 6: Results & Findings 7](#_Toc29389892)

[5.1 Group 1 7](#_Toc29389893)

[5.2 Group 2 7](#_Toc29389894)

[5.3 Performance Results 7](#_Toc29389895)

[5.4 How much did the AI learn 7](#_Toc29389896)

[5.5 Application of Learning AI in video games 7](#_Toc29389897)

[5.5 AI Profiling Results 7](#_Toc29389898)

[Chapter 7: Discussion 8](#_Toc29389899)

[Chapter 8: Conclusion 8](#_Toc29389900)

[References 8](#_Toc29389901)

[Appendences 10](#_Toc29389902)

# Terminology

# Chapter 1: Introduction

Artificial intelligence as a method of controlling characters has been around since the days of Pong. Technology and hardware for the intelligence has been rapidly growing creating new techniques, software and games. Neurological networks are an imitation of our own brain and the network that produces it and it is a close attempt to simulate an intelligence that can learn from its actions.

In the following chapters I will look at and assess the effectiveness of a learning artificial intelligence in a video game. Describe and detail the process of creating the network and assess how effective the project was as a whole.

Primary areas of interest for this project are neurological networks, the workings behind them and programming a game designed around these networks.

The project will create SERE an evasion game where the AI must hunt and capture fleeing players.

# Chapter 2: Literature Review

## 2.1 Introduction into Artificial Intelligence in video games

Artificial Intelligence is a collection of tools and algorithms that attempt to imitate intelligence Gordon, B. M. (2011). Artificial intelligence is used in video games to create non-player characters (NPCS) which interact with the world and player(s). In Real Time Strategy games such as Planetary Annihilation (Star Theory Games, 2015) the NPCs use Artificial intelligence to battle each other in a global battle for dominance, in these styles of games the player will control where the units go but battles are usually automated.

Video game, artificial intelligence uses two main techniques, goal-driven behaviour and behaviour trees. These intelligence algorithms are designed to imitate realistic patterns set out by the programmers and are not designed to be realistic. (Buckland, 2011)

In most video games artificial intelligence is meeting gamers standards with some recent modern games creating challenging AI for players to combat in their objectives in the game. Such games like Heat signature use techniques like procedural generation and goal driven behaviour to create a dynamic and challenging game.

However, the AI can only adapt to a certain calibre of player level, and if players try new and untried tactics the AI may not be able to respond, often is the case that players find the AI weakness’s and exploit that be it game mechanics or bugs within the intelligence of the NPC.

## 2.2 Introduction to Neural Networks in video games

Neural networks are the simulation of a human brain using artificial intelligence, these networks can essentially be trained and learn from their actions and can recognise patterns from data inputted into them and select certain strategies. They can also adapt more easily to player personalities and can model the player behaviour and experience. (Yannakakis & Togelius, 2018)

Neural Networks or machine learning is currently used in most fields of research to answer question which would commonly take humans years of work. Neural Networks however are relatively new to video games and hence have less research and techniques to implement them into video games.

This is largely due to the performance requirements of these artificial intelligence compared to AD-HOC techniques.

There is the issue of balancing a neurological network. Balancing issues can arise from an intelligence, new players could have a hard time adapting to a smart learning computer, players can easily find a way to abuse the training data. These are just some of the issues from a development standpoint (Boser, Sackinger, Bromley, LeCun, & Jackel, 1992)

Another reason is simply that AD-HOC AI is easier to implement, neurological networks require training and data to be built and that comes with its own issues (Yannakakis & Togelius, 2018). Simple games like Pac-man (Namco, 1980) would not benefit in any way from a neurological network.

For the purpose of this study the effectiveness of neurological networks for small simple games is void this is due to the obvious ease that an AD-HOC AI can be created for these games within a much quicker timer then a neural network. The study is focusing more on larger games with more complex inputs as these are the games that most modern developers are creating and it would be a better test to put a learning AI in a game with multiple data inputs and unpredictable variables to truly test the system.

## 2.3 Neural Networks current capabilities within video games

Neurological networks do exist in video games. Forza Horizon 4 (Turn 10 Studios, Playground Games, 2018) uses them for its driver profiles to adapt dynamically to players techniques and create more dynamic AI for their game (Forza Support, 2019). Alpha star is a modded ai system for Star craft 2 (Blizzard Entertainment, 2010), Alpha star (DeepMind, 2019) is a Deep Mind AI that learns from its matches and training data and is currently battling the professionals of the game.

Neurological networks are very rare as of 2019 in the video game industry. With the only two examples being those above. However, their capabilities for games is evident. Machine learning AI has picked up games such as chess and are very effective competitors at the game (IBM, 1997). Using this knowledge Learning AIs on a base level should be effective at strategy games, specifically playing the commander of the forces. This is because they have all the information and learn from their mistakes often completing actions in a logical order and taking routes that players would not consider. Most strategy games have some computer opponents anyway which are adequate at the game so the systems should work with a learning AI behind them, as Alpha Star (DeepMind, 2019) proves within Star Craft 2 (Blizzard Entertainment, 2010).

Neurological networks are evidently capable of being the commander, which is one component of the game, however the true test of their effectiveness would be to look at agents. Agents must process information every frame and decide their actions using ad-hoc techniques which have a set outcome, I.E. Agent sees enemy, agent shoots enemy, enemy dies, agent carries on. Those steps have been programmed out for the ad-hoc AI, but a learning AI would have to learn how to do all that (Yannakakis & Togelius, 2018). In theory the most effective way to create the learning AI would have a hybrid of the two systems. The learning AI chooses what actions it conducts, and the AD-hoc programming will complete that action.

## 2.4 Main Differences between AI and Neural Networks

The biggest and most notable draw back of a neural network is the time and resources needed for it. Hypothetically it will take one day just to train it to walk around the map, whereas the AD-HOC has already got all those systems created for it. Simply put the ad-hoc ai is cheaper and more effective to have in a video game, the game can have more agents. Training the AI requires data and input as well as time to fix any learning issues. (Yannakakis & Togelius, 2018)

Continuing this learning AIS still need to be researched for video games, they are still a relatively new technology. AD-HOC systems already have the performance, research and industry standards. Learning AI is currently only in one game in a relatively small way and it was required for them to have a server farm to run that system. (Buckland, 2011)

However, the main advantage of neural networks is the realistic nature of them, while they require allot of training behind them, they are far more complex and capable of completing tasks that ad-hoc are not due to limitations. Neural networks can expand as much as the developers allow it to and in theory could be as competent as the players in the video games. There could be a chance that these agents are used in multiplayers games to fill in the numbers that way lower player count does not impact the players enjoyment on the game.

## 2.5 Current Issues with Artificial intelligence in modern games

Artificial intelligence in games can be complex beasts which can be hard to maintain and often have bugs, even triple A titles like outer wilds had bugs with the AI getting stuck or shooting at targets that the player wouldn’t otherwise know existed. Most issues with the current AI systems are the AI can get stuck within doors, or walls with small gaps, which players can get through but the AI struggles to. The AI can also have supernatural ability to see the player, even when they would not be spotted by another player.

The biggest issue with Artificial Intelligence is within its ability to adapt for players mistakes or personalities. The players can do unknown actions and the AI would either not respond or bug out and get stuck in loops.

In video games like Arma 3 ( Bohemia Interactive, 2013) the Artificial intelligence has a reputation for having god like eyesight and can see through objects. The AI also detects people from behind even when they are making no noise, this often makes stealth in the game difficult without modifications improving the AI.

AD-HOC also has a smaller issue with only being able to perform set tasks. If the AI needs to perform a new task or do something different it will need to be told to do that. This results in the AI only being as dynamic as it is programmed to be, while it can have responses to stimuli and actions upon that, if it hasn’t been programmed to complete that task or respond to that set of inputs it won’t do anything.

## 2.6 Industry Problems with Neural networks

Neural networks are rarely incorporated within the video game industry due to problems with the technology. The first and most substantial problem is resources needed to create a neural network currently there are multiple methods to creating a neural network (Yannakakis & Togelius, 2018), this will lead to unnecessary developer research into what method they need to use. Adding to this problem with resources is the time and data needed to teach the learning AI, hypothetically each node takes 500ms of a CPU and requires 1 day to train the data for that node, the more nodes added the bigger and more expensive the AI gets on the CPU. (Yannakakis & Togelius, 2018)

The second problem with the method is the current fact that video games can incorporate dynamic AI which can be realistic and immersive without the need for learning algorithms. Games like Heat Signature (Suspicious Developments, 2017) and Outer Worlds (Obsidian Entertainment, Virtuos, 2019) already have dynamic and realistic AI responses with little issues. Developers can create gold standard AI without the amount of time it takes to create the learning AI and with the added benefit of industry standards, documentation and resources on creating a better AI. AD-HOC AI within the industry is more preferable as it is easier to implement and does not require as many resources to create it.

Even though there are processes like reinforced learning a minor issue with neural networks is not being able to guarantee what a learning AI profile does if it gets shot at there is no guaranteed way it will seek cover if its brain decides to run at the player and stab them. This can lead to issues with the AI, while this does create dynamic gameplay, such as unpredictable behaviour and often doing the wrong action at the wrong time.

# Chapter 3: Methodology

## 3.1 Time Management

To manage the time, I have chosen the Kanban method. Kanban is a method of time management for just in time manufacturing. The work is put on a board which tracks tasks needed for the artefact and essay in the stages of the development cycle. Stages are not set in stone. This is effective for the project as a certain task may suddenly need to be developed or an idea was no longer viable.

As Kanban is less structured then other methodologies there are no deadlines, working to deadlines has always been more effective for me. This avoids the pushing stuff tasks aside because they do not have a deadline and leaving it to last minute.

To manage my time effectively I have chosen to keep a tight week development cycle. If a system has taken to long it gets pushed to the stretch goal region that way I can focus on the core systems of the project.

The artefact will need to be complete within 3 months to gather primary research, this requires me to have an altered Kanban system and run the project like an extended game jam having prototype systems instead of clean created code.

## 3.2 Analysing the artefact

To analyse the artefact, I have split it into two areas: Primary research of how a group of players interact with the game and primary research of how effective the learning AI is.

For the group it will be a focus group of 20 players, from any background possible, the focus group will complete a blind test on two games. I choose to do a blind test as it is a more effective way of gathering data (Haverford College, 2015). The blind variable for these tests will be which game has the learning AI and which game has the AD-HOC. The players will get a chance to play both games for 30 minutes, the only difference will be the AI. After each session with the game the players will be given a survey to fill out asking them quantitate and qualitative questions to gather data. After this it will be revealed as to which AI was the learning and they will have one final game with the learning ai, after this taking a final survey.

To measure how effective the development of a learning AI was I am going took at two areas. The performance of the learning AI and the development process of creating a learning AI. The performance will look at CPU, GPU, RAM AND DISK usage using unities profiler in test cases with both the learning AI and the AD-HOC AI seeing if the learning AI takes to many resources. The next is to look back at the development through GitHub pushes, project diary and notes as to how the process of creating a game for a learning AI went.

## 3.3 Software

Due to past knowledge and its ease of access to prototypes I have chosen to use Unity (Unity Technologies, 2019) for the project. The ad-hoc AI is going to be created using pre-created AI systems from either Unity (Unity Technologies, 2019) or online resources. Unity (Unity Technologies, 2019) is effective at creating the systems I need due to the nature of the component system it allows me to easily create the world and systems I want for the artefact and also allows me to create more dynamic gameplay in a quicker nature then unreal. It is also easier to develop for and to debug in making ideal for a project of 3 months.

# Chapter 5: Creating the AI Systems

## 5.1 SERE

SERE is a game based on the United Kingdom Special Forces selection process, the game aims to simulate this process of evasion. The learning AI will be trained to be the people searching for the players. The AI will get input from dogs, helicopters, thermal cameras and more to help them in their search. The learning AI will learn and use searching patterns to find the players within an area and their objective is to capture the players. The players must reach an extraction point without being caught by the AI. The world is procedurally generated which adds challenge both to the AI and players, the world is roughly 10km squared and can have various biomes and areas in it including jungle, woodland, rivers, desert, urban, fields and more.

## 5.2 Designing a game for an AI that learns

1. AI needs stimulus to learn from
2. More things the ai can do the more time it can take to learn
3. The AI Needs sensors, tags and actions for it to complete tasks
4. The learning system you are using

## 5.3 Building the AD HOC AI

1. Basic Behaviour tree
2. 3 states Searching, Hunting, Combat
3. Finds and kills anything in the enemy layer.
4. Using basic steering behaviours to wander around.

## 5.4 Building the Learning AI

1. What the learning AI needed to learn
2. The vector actions
3. Reward system
4. Getting the AI to fire
5. Sensors and behaviour

## 5.4 Training the AI

1. YAML Settings
2. Unity Freezing
3. Getting the correct rewards
4. Brain 1
5. Brain 2

## Testing the AD-HOC AI

1. Searching for the enemy
2. Engaging the enemy
3. Wandering
4. Balance

## 5.6 Testing the Learning AI

* Searching for the enemy
* Shooting the enemy
* Only engaging the enemy
* Stopping the enemy from escaping
* Balance

## 5.7 Initial analysis of the effectiveness

* The learning ai is very childlike still making a few mistakes
* Currently for a full release title the learning agents would be very basic or not able to be implemented at all
* Even though its very basic the learning ai does work

## 5.8 Performance

* FPS is good and the game is not lagging even though there are multiple copies
* Learning process did not take that long and was done on a home pc
* The only thing that takes away the performance is the bullets.

# Chapter 6: Results & Findings

## 6.1 Ad-hoc AI

* Would only find the enemy if it were “lucky”
* The wandering was random and had no real goal
* Would successfully find the ai
* With more development the ai could have more behaviour trees nodes for it to be a more advanced hunter.
* The AI can still hunt effectively.

## 6.2 Learning agent

* Most of the time it w

## 6.3 Comparing the two AI

## 6.4 Performance Results

## 6.5 How much did the AI learn

## 6.6 Application of Learning AI in video games

* Hive mind and strategy games / Commanders
* Standard AI agents that learn
* NPCS
* Puzzle making
* Other
  + Asset creation
  + Level generation
  + Auto translating voices and text for multiplayer games

# Chapter 7: Discussion

# Chapter 8: Reflection

# Chapter 9: Conclusion

# References

Bohemia Interactive. (2013, September 12). ARMA 3. *ARMA 3*. Prague, Czechia, Czech Republic: Bohemia Interactive.

Berges, V.-P., & Meuleau, N. (2019). Unity AI and Machine Learning Tools for Behavior Creation. *Unity AI and Machine Learning Tools for Behavior Creation*. GDC. Retrieved from https://www.gdcvault.com/play/1026172/Unity-AI-and-Machine-Learning

Biran, Y., Finch, C., Khim, S., O'Brien, R., & Subramanian, S. (2019). Getting Started with Machine Learning and Artificial Intelligence. *Getting Started with Machine Learning and Artificial Intelligence*. GDC. Retrieved from https://www.gdcvault.com/play/1026138/Getting-Started-with-Machine-Learning

Blizzard Entertainment. (2010, July 27). StarCraft II: Wings of Liberty. *StarCraft II: Wings of Liberty*. Blizzard Entertainment.

Boser, B. E., Sackinger, E., Bromley, J., LeCun, Y., & Jackel, L. D. (1992, Febuary). *Hardware Requirements for neural network pattern classifiers*. Retrieved from AT&T Bell Laboratories: http://yann.lecun.com/exdb/publis/pdf/boser-92a.pdf

Briskman, D. (2018). Reinforced Machine Learning: Leveraging the Power of Intelligence for Gaming . GDC. Retrieved from https://www.gdcvault.com/browse/gdc-18/play/1024835

Buckland, M. (2011). *Programming Game AI By Example.* Sudbury: Jones and Bartlett Publishers, Inc.

Catanzaro, B. (2017). Deep Learning for Game Developers. *Deep Learning for Game Developers*. GDC. Retrieved from https://www.gdcvault.com/play/1024260/Deep-Learning-for-Game-Developers

DeepMind. (2019, January 24). *AlphaStar: Mastering the Real-Time Strategy Game StarCraft II*. Retrieved from Deep Mind: https://deepmind.com/blog/article/alphastar-mastering-real-time-strategy-game-starcraft-ii

Forza Support. (2019). *Drivatars*. Retrieved from Forza Support: https://support.forzamotorsport.net/hc/en-us/articles/360005302934-Drivatars

Gordon, B. M. (2011). *Artificial intelligence approaches, tools and applications.* New York: Nova Science Publications, Inc.

Hammoudi, S., Okumura, J., & Tanaka, I. (2019). Applying AI in Games with DeNA. GDC. Retrieved from https://www.gdcvault.com/play/1026157/Applying-AI-in-Games-with

Haverford College. (2015). *Science Exposed*. Retrieved from Psych 2015: http://ds-wordpress.haverford.edu/psych2015/projects/chapter/blind-testing/

IBM. (1997, May 11). Deep Blue. *Deep Blue*. IBM.

Luo, J. J. (2019, May 22). *An Exploration of Neural Networks Playing Video Games*. Retrieved from Towards Data Science: https://towardsdatascience.com/an-exploration-of-neural-networks-playing-video-games-3910dcee8e4a

Macri, D. (2017, November 19). *An Introduction to Neural Networks With an Application to Games*. Retrieved from Intel Developer Zone: https://software.intel.com/en-us/articles/an-introduction-to-neural-networks-with-an-application-to-games

Namco. (1980, May 22). Pac-man. *Pac-man*. Namco.

Nicholson, C. (n.d.). *A Beginner's Guide to Neural Networks and Deep Learning*. Retrieved from Pathmind: https://pathmind.com/wiki/neural-network#apllyingLossFunction

Nvidia. (n.d.). *Deep Learning*. Retrieved from nvidia Developer: https://developer.nvidia.com/deep-learning

Obsidian Entertainment, Virtuos. (2019, October 25). The Outer Worlds. *The Outer Worlds*. Private Division, Take-Two Interactive.

Pomerleau, D. A. (1991, March). *Efficient Training of Artificial Neural Networks for Autonomous Navigation.* MITP.

Rabin, S. (2017). *Game AI Pro 3: Collected Wisdom of Game AI Professionals.* Boca Raton: Taylor & Francis.

Reynolds. (2000, June 2). *Research*. Retrieved from Reynolds: https://www.red3d.com/research.html

SINGH, H. (2019, February 24). *Everything you Need to Know About Hardware Requirements for Machine Learning*. Retrieved from EInfochips: https://www.einfochips.com/blog/everything-you-need-to-know-about-hardware-requirements-for-machine-learning/

Skinner, G., & Walmsley, T. (2019). *Artificial Intelligence and Deep Learning in Video Games A Brief Review.* IEEE.

Star Theory Games. (2015, August 18). Planetary Annihilation Titans. *Planetary Annihilation Titans*. Kirkland, Washington, United States: Planetary Annihilation Inc.

Suspicious Developments. (2017, September 21). Heat Signature. *Heat Signature*. Suspicious Developments.

The AlphaStar Team. (2019, October 30). *AlphaStar: Grandmaster level in StarCraft II using multi-agent reinforcement learning*. Retrieved from Deep Mind: https://www.deepmind.com/blog/article/AlphaStar-Grandmaster-level-in-StarCraft-II-using-multi-agent-reinforcement-learning

Turn 10 Studios, Playground Games. (2018, September 28). Forza Horizon 4. *Forza Horizon 4*. Redmond, Washington, United States of America: Xbox Studios.

Unity. (2018). *Get to grips on the latest AI power in Unity*. Retrieved from Unity: https://unity3d.com/how-to/unity-machine-learning-agents

Unity Technologies. (2019, December 11). Unity. *Unity*. Unity Technologies.

Yannakakis, G. N., & Togelius, J. (2018). *Artificial intelligence and games.* Cham, Switzerland: Springer.

# Appendences

## 1 – Survive, Evade, Resist, Extract Design idea

SERE

### Quick Fire

* SERE is a game based on the United Kingdom Special Forces selection process, the game aims to simulate this process of evading a force hunting for you.
* The game puts you in control of the hunted, with your fellow players you must escape to a point on your map.
* 5 minute head start.
* Players leave behind scent smells for dog and thermal footprints as well as other methods of tracking them down including audio
* SERE has two games as it is designed to be tested, the games are called game 1 and game 2 so not to give away which AI system is controlling them
* The players will have 30 minutes with each game.
* The players are given a weapon to defend themselves with, However the weapon is loud and will attract more attention.
* The player can win in 2 ways, kill all the enemies or reach the extraction point.

### Brief:

·        SERE is a game based on the United Kingdom Special Forces selection process, the game aims to simulate this process of evasion. The learning AI will be trained to be the people searching for the players. The AI will get input from dogs, helicopters, thermal cameras and more to help them in their search. The learning AI will learn and use searching patterns to find the players within an area and their objective is to capture the players. The players must reach an extraction point without being caught by the AI. The world is procedurally generated which adds challenge both to the AI and players, the world is roughly 10km squared and can have various biomes and areas in it including jungle, woodland, rivers, desert, urban, fields and more.

### Units

1. Rifleman
2. Tracker
3. Dog
4. Helicopter
5. Drone
6. Team Leader
7. Thermal Camera Operator
8. Sniper
9. Squad Leader
10. Platoon Leader
11. Company Commander

### AD-HOC AI (Game 1)

The AD-HOC AI has the same information given to them as the learning AI. The only difference between the two systems is the AD-HOC ai are using a goal driven behavior tree system to choose their actions. The AIS in this system always passes the information up the chain of command.

### Learning AI (Game 2)

·        The learning AIS information is received through a system called filter. The Agent knows what they see, and what their “senses” perceive, and they call out any important information to the person leading their team. This team leader through their learning decided if that information needs to be passed up the chain of command or not. If they call it out on the radio, every other team leader knows this information, team leaders then choose if they filter that information back down through their learning. This is a basic way of explaining it

o   Ok so the teams are organised how the military would have them, as I am simulating this I have given the learning AI profiles based on their rank, the higher their rank the more leadership weighed their decisions are.

o   So you have a team, squad, platoon. The company commander is the stop point of all commands. He is the most sophisticated learning AI as he is making the decisions of where the platoons are actually going and through game systems receives the most information.

o   Each agent in this will be learning agents, this will test how effective the process is.

·        There is learning agent profiles, a method I created to simulate dynamic behaviour while still being predictable in the way I intend it to be

o   Rifleman

o   Dog

o   Helicopter

o   Thermal Camera Operator

o   Sniper

o   Team Leader

o   Squad Leader

o   Platoon Leader

o   Company Commander

## 2 - SERE Pitch Document

SERE: Survive, Evade, Resist, Extract. Game Design Document

### High Concept

Players must escape from a hunter force and reach an extraction point, there are multiple ways to escape or be captured. Players can choose to fight and kill all the hunters or escape without a trace.

### Unique Selling Points

1. Learning AI
2. Slow Gun play
3. Changing level design
4. Focused game play

### Game Overview

In SERE you play as an armed forces member trying to get into the United Kingdom Special Forces. You are at the final stage and are being challenged with the final SERE trial. Your objective is to escape the hunters and use any skills, equipment or terrain to your advantage and learn how to avoid an enemy force actively hunting you. The game is open world with a border at the edge of the world to stop the players and ai from accidently leaving the area. The game is primarily stealth but players can run and gun.

### Game Objective

The primary objective of the game is to survive and escape without the enemy force knowing you were there. The players must reach a given extraction point usually a reasonable distance away to add the challenge.

### Game Genre

Stealth Action - Adventure with open world elements

### Target Audience

The target audience of SERE is 17+ with it aiming towards audiences who like stealth games and wish to be challenged.

### Controls

W: Go Forwards

A: Go Back

S: Go Left

D: Go Right

Mouse: Aim

Left Click: Shoot

Right click: Scope in/zoom

Space: Jump/climb

Shift: Sprint

Z Prone/Stand up

X Crouch/Stand up

Control: Ultra Slow Movement

M: Map

### Gameplay mechanics

#### Learning AI

The learning AI system will be a hybrid system custom made for the game. AD-HOC will be the body and complete the actions that the learning AI brain chooses to do, this way the learning AI will not have to learn steering behaviours. The brain will have information stored with what it knows acting like an active memory which includes details told to them by squad members and what they have seen. They take this information and choose an action based on the desired strategy. Each learning AI will be a profile assigned to a role that way it only trains and learns what it needs to know. Each AI will be slightly different in how it performs an action but they should all roughly perform in the same light. The learning AI will also learn what the players do and start to adapt against them.

#### Senses

The AI will have simulated senses with being able to hear, see and smell. Some roles like dogs have a heightened sense of smell and can track the players using this sense. Each role has different sense which tell its brain more or less information in that area and then pass up that information.

#### Filter (Communications)

The AI uses a system which I am calling filter, essentially the AI chooses to pass information up or down the chain of command. Each role has a radio they are on and can hear for example squad leaders are on their platoon radio and team radio and can hear and pass information on both. This system is used to allow the player to break up the radio communication and talking and also to make sure that the AIs are not always all knowing of each others action they have to choose to pass this information to each other. The short way it works is an AI senses something and then it triggers a response and if the brain believes it should communicate that up to its team leader they will.

Example of how this works :

1. Dog smells a player more clearly and hears a branch break
2. Dog decides to bark as its brain thinks the player is close
3. Handler hears the dog bark and knows that this means a player is nearby
4. Handler tells their team leader that their dog may of found something
5. Team leader passes onto their squad leader
6. Squad leader calls for their squad to regroup on the handlers position and perform a sweep of the area.
7. Squad leader informs their platoon leader of this action.
8. Platoon leader calls for the other 2 squads to quickly move to this location and start sweeping in a pattern 500 meters away
9. Platoon leader tells their company commander of a player being near their platoon and tells the company commander the location
10. Company commander will inform all units of the possible location of the player and then their brains will make decisions on what to do

#### Sound

The game will have basic sounds like branch breaking, dogs barking, helicopter engines, radios and ai “talking” to each other and gunshots that way the player and AI can all hear these sounds. Sound quality does not need to be paramount and can be done by getting sample sounds.

#### Basic Combat

The game will have basic projectile combat where shoot a small ball at the enemy and they shoot at you with the same the small ball deals 20 damage no matter where it hits. There are no “guns” or weapons in the inventory instead everyone has the ability to shoot at a set fire rate and all have the same weapon.

#### Procedural map

The map will be procedural generated in a 10KM by 10KM system, first a map theme is chosen from Desert, Jungle, Forest or marshland. This choice gives the height map generator parameters to generate a random terrain for the players to walk across. Then the features like rivers,  trees, rocks and more are placed. A rough texture and map are then created.

#### AI Roles/Profiles

Each AI will have a role which then gives them their brain. The roles or profiles are just another name for the AIS brain which chooses actions based on their role, so a helicopter would only have actions and choices based on what a helicopter can do so they wont need to be able to fire or have any code for shooting at players . Roles will change what the AI is able to do. The current planned roles are

* Helicopter
* Dog
* Rifleman
* Thermal Camera Operator
* Team Leader / Squad Leader
* Platoon Commander
* Company Commander

### Gameflow

The game is fully singleplayer with the learning AI already being trained and having the necessary data for it to run, it can still learn from its actions and create new responses towards the players actions. The game is open world and has the ability for the players to hide from the hunting force and use the terrain to their advantage. The game's progress is linked to how competent the player is and in theory the player may never reach the extraction point but also never be found.

### Technology

The game will use unity due to how quick and easy it is to create a prototype game and the ease of systems in unity, with the added benefit of unity already has open source machine learning agents I can learn and adapt for the project. Other technology includes any AD-HOC system I need for the game to work.