NEWCASTLE UNIVERSITY

**Investigating AI approaches and how they challenge resolutions: State AI vs Machine Learning**

**Chloe Sunter | 180273436**

MComp Computer Science

May 2022

Supervisor: Dr Giacomo Bergami

Word Count: 1234

Abstract

Declaration

“I declare that this dissertation represents my own work except where otherwise stated.”

Acknowledgements

Firstly, I would like to thank my supervisor, Dr Giacomo Bergami, for his guidance and support through each stage of the process.

Dr Giacomo Bergami

Dr Rich Davison???

Dr Graham Morgan

Dr Gary Ushaw

Contents

[Abstract 2](#_Toc102871690)

[Declaration 2](#_Toc102871691)

[Acknowledgements 3](#_Toc102871692)

[1 - Introduction 6](#_Toc102871693)

[1.1 - Dissertation Structure 6](#_Toc102871694)

[1.2 - Definition of Problem 6](#_Toc102871695)

[1.3 - Hypothesis 6](#_Toc102871696)

[2 - Aims & Objectives 7](#_Toc102871697)

[3 - Background Review 8](#_Toc102871698)

[3.1 - Hardpoint 8](#_Toc102871699)

[3.1.1 - Rules 8](#_Toc102871700)

[3.1.2 - Examples from the real world 9](#_Toc102871701)

[3.2 - AI 9](#_Toc102871702)

[Overview 9](#_Toc102871703)

[State AI 9](#_Toc102871704)

[Finite state, heiracly finite state, probalistic finite 9](#_Toc102871705)

[Machine Learning 9](#_Toc102871706)

[Algorithms, compare etc, Deep Reinforcement, Imitation Learning 9](#_Toc102871707)

[3.3 - Tools & Technologies 9](#_Toc102871708)

[Software 9](#_Toc102871709)

[Unity, compare other options perhaps. 9](#_Toc102871710)

[Mlagents, do a lot of research into this. What alorithms it uses etc 9](#_Toc102871711)

[Hardware 9](#_Toc102871712)

[Kind of like system requirements. Obvs tho, I don’t match those. Maybe try to find some graphs /studies that show how better equipment can affect the training times etc 9](#_Toc102871713)

[3.4 - Similar Projects 9](#_Toc102871714)

[4 - Project Development 10](#_Toc102871715)

[4.1 - System Setup 10](#_Toc102871716)

[4.1.1 - Hardware 10](#_Toc102871717)

[4.1.2 - Version Control 10](#_Toc102871718)

[4.1.3 - Software 10](#_Toc102871719)

[4.2 - Design 11](#_Toc102871720)

[4.2.1 - Map Design 11](#_Toc102871721)

[4.2.2 - Gameplay Rules 11](#_Toc102871722)

[4.2.3 - State Machine 11](#_Toc102871723)

[4.2.4 - Deep Reinforcement setup? 11](#_Toc102871724)

[4.3 - Implementation 11](#_Toc102871725)

[4.4 - Testing 11](#_Toc102871726)

[5 - Results & Evaluation 12](#_Toc102871727)

[6 - Conclusion 13](#_Toc102871728)

[7 - References 14](#_Toc102871729)

[8 - Appendices 15](#_Toc102871730)

1 - Introduction

1.1 - Dissertation Structure

1.2 - Definition of Problem

1.3 - Hypothesis

2 - Aims & Objectives

3 - Background Review

3.1 - Hardpoint

Hardpoint is a game style that is similar to Capture the Flag and has been implemented as a main game mode in many popular online multiplayer games. Whilst this style of game is commonly known as “Hardpoint”, its name varies across different franchises. Other names include: “Stronghold” (Halo), “Headquarters (Call of Duty), “Hardpoint Domination” (Titanfall) to name a few. Each franchise has modified the game to fit their own style, however the foundation stays the same.

3.1.1 - Rules

Overview

Two teams play against each other on a map that has designated areas (hardpoints) that reward points when players enter them. The minimum amount of hardpoints is 3, with one of these hardpoints being equal distance from both team’s respawn areas. The aim of the game is to beat the opposing team by reaching the total score before them (for example, 100points). Points can be earnt by capturing the hardpoint from the other team and for defending the hardpoint from the opposing team.

Capture

A player can only capture a hardpoint that is either owned by the enemy or has yet to be captured by anyone for the first time in a game. To capture it, a minimum of one player must enter the hardpoint and stay within the area for a small period of time (for example, 3seconds). If an enemy enters the area during this time, the hardpoint will become congested. Once captured, the team gains a large portion of points (for example, +15points).

Defend

A player can only defend a hardpoint that is owned by its own team. To defend it, a minimum of one player must enter and stay within the area for an extended period of time (for example, 10seconds). If an enemy enters the area during this time, the hardpoint will become congested. Once defended, the team gains a small portion of points (for example, +5points). This will repeat for as long as a player is within the hardpoint and it belongs to said player’s team.

Congested

If players from both teams are present in the hardpoint at the same time, it becomes “congested” and no points are rewarded whilst the hardpoint is in this state. To exit this state, a team must remove all enemy players from the area. Once complete, the hardpoint will go back to the state it was beforehand and normal gameplay continues.

Kills

Players can kill other players, however, this does not affect the score. Respawns are enabled.

3.1.2 - Examples from the real world

3.2 - AI

Overview

State AI

Finite state, heiracly finite state, probalistic finite

Machine Learning

Algorithms, compare etc, Deep Reinforcement, Imitation Learning

3.3 - Tools & Technologies

Software

Unity, compare other options perhaps.

Mlagents, do a lot of research into this. What alorithms it uses etc

Hardware

Kind of like system requirements. Obvs tho, I don’t match those. Maybe try to find some graphs /studies that show how better equipment can affect the training times etc

Please see <> for specific versions used in this project.

3.4 - Similar Projects

4 - Project Development

4.1 - System Setup

asfasfd

4.1.1 - Hardware

|  |  |
| --- | --- |
| Device | Laptop |
| Processor | Intel® Core™ i5-7200U CPU @ 2.50GHz, 2713 MHz |
| Core(s) | 2 |
| Logical Processor(s) | 4 |
| Operating System | Microsoft Windows 10 Home |
| Memory *(RAM)* | 8GB |
| Graphics Processing Unit *(GPU)* | Intel® HD Graphics 620 |

4.1.2 - Version Control

|  |  |
| --- | --- |
| Git | version.2.21.0.windows.1 |
| Github Desktop |  |
| Gitbash |  |

4.1.3 - Software

Editors

|  |  |
| --- | --- |
| Unity | version.2021.1.21f1 |
| Vs Code | version.1.67.0 |
| Microsoft Excel | version.2204 |

Languages

|  |  |
| --- | --- |
| Python | version.3.9.9 |
| C# | n/a |
|  |  |

Toolkits, Frameworks, Libraries & Packages

|  |  |
| --- | --- |
| ML-Agents Toolkit | release.19 |
| com.unity.ml-agents | version.2.2.1-exp.1 |
| com.unity.ml-agents.extensions | version.0.6.1-preview |
| ml-agents | version.0.28.0 |
| ml-agents-envs | version.0.28.0 |
| gym-unity | version.0.28.0 |
| Communicator | version.1.5.0 |
| Probuilder |  |
| Tensorflow | version.2.8.0 |
| Tensorboard | version.2.8.0 |
| Torch | version.1.7.1+cu110 |
| Pip | version.22.0.4 |
|  |  |

*Please note: This is not a full list of all packages included. It only includes the most prominent ones which may be mentioned throughout this dissertation.*

4.2 - Design

4.2.1 - Map Design

4.2.2 - Gameplay Rules

I have implemented the basic version of the Hardpoint game mode outlined in <>. Below is a list of parameters that I have used to run the game. These will stay the same for all playthroughs so as to not affect the result comparisons.

Total number of teams: 2

Total number of players per team: 4

Total number of Hardpoints: 3

Score needed to win: 100

Time (seconds) it takes to capture: 3

Time (seconds) it takes to defend: 10

Points earnt for capturing: 15

Points earnt for defending: 5

Points earnt for killing an enemy: 0

Respawns enabled: Yes

4.2.3 - State Machine

4.2.4 - Deep Reinforcement setup?

4.3 - Implementation

4.4 - Testing

5 - Results & Evaluation

6 - Conclusion

7 - References

8 - Appendices