Player Emulation in Video Games using Artificial Intelligence

B.Sc. (HONS) Computing with Games Development

Supervisor: Robert Sheehy

Student Number: T00200674

Student: Ben Lenihan

Munster Technological University, Dromthacker, Tralee, Co. Kerry

# Abstract

Artificial intelligences are computer systems designed to replicate human problem-solving abilities. They are widely used in many forms and across many fields such as video games, where they are used to control non-player characters, and as a way of automating more difficult but repetitive tasks in many technical areas.

The aim of this project is to create an artificial intelligence to replace human interaction in a video game. The game chosen for this was a Japanese mobile game by the name of “Fate/Grand Order”. The artificial intelligence model that was chosen was a decision tree with the addition of template matching for use in reading the game screen.

The proof concept in the decision tree proved mostly capable of creating simple strategies to win battles in the game and to some extent was able to create more advanced strategies. However, the template matching was unable to accurately find objects on screen. This meant the decision tree was unable to be fully utilized.

1. Table of Contents

[Abstract 2](#_Toc88614959)

[1 Table of Contents 2](#_Toc88614960)

[1 Introduction 3](#_Toc88614961)

[2 Artificial Intelligence 3](#_Toc88614962)

[2.1 Introduction 3](#_Toc88614963)

[2.2 Machine Learning 3](#_Toc88614964)

[3 Algorithms 4](#_Toc88614965)

[3.1 Decision Trees 4](#_Toc88614966)

[4 Technologies 5](#_Toc88614967)

[4.1 Android Emulators 5](#_Toc88614968)

[4.1.1 Bluestacks 5](#_Toc88614969)

[4.1.2 LDPlayer 5](#_Toc88614970)

[4.1.3 NoxPlayer 6](#_Toc88614971)

[4.2 Evaluation of Android Emulators 6](#_Toc88614972)

[4.3 Programming Languages 8](#_Toc88614973)

[5 Methodology 8](#_Toc88614974)

[5.1 Research Undertaken 8](#_Toc88614975)

[5.2 Research Question 8](#_Toc88614976)

[5.3 Proposed Project Implementation 9](#_Toc88614977)

[5.4 System Design 9](#_Toc88614978)

[5.5 Prototype 9](#_Toc88614979)

[6 Implementation 9](#_Toc88614980)

[6.1 Sprints 9](#_Toc88614981)

[6.1.1 Sprint 1 9](#_Toc88614982)

[7 Findings & Conclusions 10](#_Toc88614983)

[8 References 10](#_Toc88614984)

# Introduction

Artificial intelligence is a field of computing which is focused on replicating human interaction with computer programming. Artificial intelligence (AI) is widely used in many areas such as video games, virtual assistants, image analysis, data analysis, search engines and robots.

AI is used to lessen the burden of human resources where possible. While it would be highly impractical for a human of sift through large amounts of data as it would take a great deal of time. An AI can potentially do this in a fraction of the time.

Video games often feature a lot of problem solving. This makes them a great area to test the capabilities of AI. This project aims to use an AI system on a game to test its capabilities.

# Artificial Intelligence

## Introduction

Artificial Intelligence is a branch of computer science concerned with building programs that can perform tasks that would, under normal circumstances, require human intelligence. “It is the science and engineering of making especially intelligent computer systems.” (Mccarthy, 2004)

The idea of Artificial Intelligence can be traced back as far as the 1950’s with Alan Turing’s work “Computing Machinery and Intelligence”. In this paper Turing asks the question “Can machines think?”(Turing, 1950). He then establishes out the “Turing Test”. This is a test in which a human interrogator is supposedly able to distinguish between a machine and a human. Developments since then have already allowed Artificial Intelligence to surpass humans in some areas. In 2015 Google’s AI AlphaGo played the European Go champion Fan Hui.(Stanek, 2021)

Artificial Intelligence is utilized in many areas such as “assistants” in the form of Apple’s Siri, in games for non-player characters, self-driving cars and the AlphaGo AI that beat Fan Hui.(European Parliament, 2021)

## Types of Artificial Intelligences

Reactive Machines: These are the simplest type of AI. They react to an input with an output, giving the same output every time they are given the same input. These store none of the inputs and perform no learning.

Limited Memory: These AI use prior data to improve future predictions. These learn overtime improving through cycles.

The final two types, Theory of Mind and Self-Aware, are fully hypothetical. These AI would be incredibly human-like.

Theory of Mind: These AI would be able interact with thoughts and emotions of humans.

Self-Aware: These AI would be a fully self-aware being, capable of their own thoughts and actions.(Johnson, 2020)

## Machine Learning

Machine learning is a sub-branch of Artificial Intelligence focusing on the use of algorithms and data to replicate the way humans learn.

UC Berkeley describe a typical machine learning algorithm as follows:

1. **A decision process:** A recipe of calculations or other steps that takes in the data and returns a “guess” at the kind of pattern in the data your algorithm is looking to find.
2. **An error function:** A method of measuring how good the guess was by comparing it to known examples (when they are available).
3. **An updating or optimization process:** Where the algorithm looks at the miss and then updates how the decision process comes to the final decision so that the next time the miss won’t be as great.(Tamir, 2021)

# Algorithms

## Introduction

An algorithm is a procedure created to solve a well-defined computational problem. Algorithms are fundamental component of computer science used in all branches such as artificial intelligence, databased, operating systems and many more. Creating an algorithm requires an understanding of the problem, the available solutions and “what it means for the algorithm to be correct” in that it efficiently solves the whole problem.(Britannica, 2022a)

## Decision Trees

A decision tree is an algorithm used for machine learning. A decision tree starts at one point (called a node) and branches into at least two directions, each branch offering different outcomes. Decision trees consist of three types of nodes.

* Decision nodes: These represent a decision to be made by the system.
* Chance nodes: These represent a probability of what will happen.
* End nodes: Representing an outcome.

These nodes are connected by branches. These nodes and branches are reusable and can be used in any number of combinations to create more complex trees. The first node is called the “root node” this, while the final nodes, representing outcomes, are called the leaf nodes. The nodes between these are called internal nodes, these represent decisions or chances depending on the tree’s purpose. Sometimes these trees can put too much emphasis on irrelevant data. In these instances, a process named pruning can be used. In this process unnecessary data is removed.(Hillier, 2021)

Diagram

Description automatically generated

Figure 3.1. - Simple Decision Tree using chance nodes(Hillier, 2021)

## Advantages and Disadvantages of Decision Trees

Using decision trees creates many advantages:

* Decision trees are simple to understand and can be easily visualised.
* Data does not need to be specially prepared for use.
* Decision trees can use both numerical and categorical data.
* Decision trees can handle multi-output problems

However, there are drawbacks to decision trees as well:

* Decision trees can become overly complex if the data is not generalised well. This can be avoided by pruning the tree ridding it of unnecessary items or by setting a maximum depth for the tree.
* Small variations can cause vastly different trees making them unstable in certain circumstances.
* The predictions of a tree are not smooth, they are approximations and are therefore not good at extrapolation

(Scikit-learn, 2022)

## Neural Networks

A neural network is a subset of machine learning inspired by the processes of the human brain. Neural networks attempt to mimic the ways in which biological neurons signal one another. Neural networks are made up of layers of nodes, these are divided into an input layer, one or multiple hidden layers and an output layer. The nodes connect to each other and have associated weights and threshold values. If a node is above the threshold value, it activates sending data to the next layer.

Neural networks train the data to learn and improve their own accuracy overtime. Once a network has been trained to be sufficiently accurate it enables a high velocity of data classification to be done. These algorithms are often used in speech and image recognition as well as in Google’s own search algorithm.(IBM, 2022)

Diagram

Description automatically generated

Figure 3.2 – Picture displaying a representation of the nodes of a neural network (IBM, 2022)

## Advantages and Disadvantages of Neural Networks

Advantages of a neural network

* Neural Networks are capable of dealing with data which has errors
* Is used for fast evaluation of a learned function
* Can be trained for long periods of time

Disadvantages of neural networks

* Can be very hardware intensive, requiring parallel processing power
* Functions of the network can be unexplained, reducing trust in them
* Structure of the network can require trial and error

(Asquero, 2022)

## Template Matching

Template matching is a digital image processing technique to find a section within an image that is similar to a given template image. This is done through comparing pixel values of the source image one by one to the template image. Images are often converted to greyscale to speed up the process as it won’t need to compare colours of the source and template images.(Caubalejo, 2021)

# Video Games

## Brief History of Video Games

The earliest video games date back to the 1950’s with William Higginbotham’s Tennis for Two. Built for an exhibition in Brookhaven National Laboratory in 1958 Tennis for Two is one of the earliest video games.(Brookhaven National Laboratory, 2022)

The first widely available commercial game was Pong released in 1972 by Atari. Pong, similarly, to Tennis for Two, was based on tennis, more specifically table tennis or Ping-Pong. By the end of 1972 more than 8,000 Pong Arcade machines had been sold and in 1975 the Atari console was released with Pong being among the first games.(Britannica, 2022b)

In the 1980’s many popular games were created such as Pac-Man, Donkey Kong and Tetris. Despite these well-known popular hits being released at the time the early 80s saw a massive crash in the industry. The number of games being produced vastly outpaced the market demand. Many low-quality games were being developed, which in turn lowered demand even further. This crash in the market saw industry giants crippled such as Atari who lost $356 million. While this crash greatly affected North American companies, Japanese companies, such as Nintendo, Sega and Sony, were largely unaffected. In 1985 the Nintendo Entertainment System (NES) was released in western nations and with it a renewed interest in the console market which continued to build over time. Many of these companies remain major players in the current video game market, particularly Nintendo and Sony.(Kersley, 2021; Musuem of Play, 2022)

## Video Game Emulation

Video game emulators are software which attempts to allow for computer to emulate the software of various devices such as consoles or mobile devices allowing for games previously only available on these systems. These emulators are often used to play games for consoles which are no longer produced such as PlayStation 2, old Nintendo systems and many more. Emulators for mobile devices have also been created, such as Bluestacks for Android, allowing mobile applications and games to be run on personal computers. While these systems in themselves are usually legal and playing officially purchased games on them is legal they are often used to play illegal pirated games.

# Android Emulators

An Android emulator is a software which simulates the Android operating system for mobile devices on a desktop computer. This allows applications created for Android to be run on another system such as Windows. The emulators being examined here are:

* Bluestacks
* LDPlayer
* NoxPLayer
* MEmu Play

## Bluestacks

Bluestacks is among the most popular Android emulators, claiming more than 500 million users. It features a large library of compatible Android games and is constantly growing its library. It contains many features such as keyboard control, multi-instances, allowing multiple instances of the same application to be run at the same time, and cloud-based platform usage alongside its more traditional desktop application(Bluestacks, 2021)

## LDPlayer

LDPlayer is a popular Android emulator with more than 4 million daily users. It claims to support more than 1 million games. It supports features custom keyboard controls and multi-instance allowing the same app to be opened and used at the same time.(LDPlayer, 2021)

## NoxPlayer

NoxPlayer is a popular Android emulator. Claiming to have over 150 million users across 150 countries. It supports common features such as keyboard control mapping and multi-instances of applications.(NoxPlayer, 2021)

## MEmu Play

MEmu Play is another popular Android emulator. It reports over 100 million downloads and supports keyboard mapping and multiple instances.(Microvirt, 2021b; 2021a)

## Evaluation of Android Emulators

These emulators are broadly similar in their abilities, making a decision between them difficult. However, benchmark tests done by Bluestacks show a number of differences in how resource intensive these emulators are compared to their peers. As these benchmark tests are done by Bluestacks themselves it is possible they are not entirely reliable. These benchmarks show Bluestacks as being the least resources intensive when compared to other common emulators.

# Methodology

## Research Review

Artificial Intelligence has been used to play many kinds of game replacing player input. Digital board such as Chess and Go, well as games like the Mario series.

## Research Question

An evaluation into the use of an artificial intelligence to replicate a human player in a mobile game.

## Proposed Project Implementation

This aim of this project is to design and build an artificial intelligence to play a mobile game, Fate/Grand Order. This will be done using a decision tree algorithm in Python and an android emulator in the form of Bluestacks. This is intended to investigate possible ways to automate aspects of games, replacing human input.

## Prototype

|  |  |  |
| --- | --- | --- |
| Prototype | Start Date | Finish Date |
| 1 | 12/11/2021 | 12/11/2021 |

|  |  |  |
| --- | --- | --- |
| Task Number | Details | Status |
| 1 | Setup Android emulator | Complete |
| 2 | Create Fate/Grand Order profile. | Complete |

|  |  |  |
| --- | --- | --- |
| Prototype | Start Date | Finish Date |
| 2 | 12/11/2021 | //2022 |

|  |  |  |
| --- | --- | --- |
| Task Number | Details | Status |
| 1 | Setup Python Environment | Complete |
| 2 | Create representation of Fate/Grand Order’s board using python. | Complete |
| 3 | Create | Complete |
| 4 | Create representation of Fate/Grand Order’s battle system using python. | Complete |
| 5 | Create a Decision tree prototype | Complete |

# Implementation

## Sprints

### Sprint 1

|  |  |  |  |
| --- | --- | --- | --- |
| Sprint Number | Sprint Name | Start Date | Finish Date |
| 1 | Prepare Game Board | 17/01/22 | 22/01/22 |

|  |  |  |
| --- | --- | --- |
| Task Number | Details | Status |
| 1 | Setup Google Colab notebook | Completed |
| 2 | Create a Player Character class | Completed |
| 3 | Create an Enemy character class | Completed |
| 4 | Create a Card class | Completed |
| 5 | Create a Board class | Completed |

A Google Colab/Jupyter notebook was created. Classes were then created to mimic player and non-player characters from the game, Fate/Grand Order. A class was created to represent the player character owned attack cards. Finally, a Board class was created to represent the in-game board, storing current player and non-player characters

### Sprint 2

|  |  |  |  |
| --- | --- | --- | --- |
| Sprint Number | Sprint Name | Start Date | Finish Date |
| 1 | Prepare Search Tree | 24/01/22 | 12/02/22 |

|  |  |  |
| --- | --- | --- |
| Task Number | Details | Status |
| 2 | Create a Node class to store information about each turn in the game. | Completed |
| 3 | Create a method to create children, Nodes showing each possible sequence of events, for the first node. | Completed |
| 4 | Create a method for making children for each subsequent child Node of the first for three turns | Completed |

A search tree containing all possible moves player characters can take for the next three turns. This search tree takes all player owned attack cards, going through every combination of all fifteen cards picking out five. From these five, it creates every combination of three attacks the player could take that turn. It repeats this process again for the next two turns, for a total of three turns.

### Sprint 3

|  |  |  |  |
| --- | --- | --- | --- |
| Sprint Number | Sprint Name | Start Date | Finish Date |
| 3 | Prepare Decision Algorithm | 14/02/22 | 25/03/22 |

|  |  |  |
| --- | --- | --- |
| Task Number | Details | Status |
| 1 | Create method to calculate the damage each move will do | Completed |
| 2 | Create a method which finds the move which will do the maximum damage for each move | Completed |
| 3 | Utilise these to find the optimal for each possible path for each | Completed |

This algorithm searches through the cards for the combination of cards which will deal the maximum damage to the opponent. This is achieved by creating every possible combination of cards. These cards are then evaluated by the total damage the combination would do against the opponent. The sets are then sorted from most to least damage. Finally, the set with the highest damage is selected.

### Sprint 4

|  |  |  |  |
| --- | --- | --- | --- |
| Sprint Number | Sprint Name | Start Date | Finish Date |
| 4 | Template Matching Test | 28/03/22 | 08/04/22 |

|  |  |  |
| --- | --- | --- |
| Task Number | Details | Status |
| 1 | Import opencv2 | Completed |
| 2 | Choose images to test template matching methods | Completed |
| 3 | Create template images from selected images | Completed |
| 4 | Perform template matching with all methods | Completed |
| 5 | Select method which gives the most consistent and accurate results | Completed |

A set of images is selected. For each image a small section is copied, such as a face or other recognisable feature, into its own separate image. This image is then compared against the previous larger image it was taken from. This is done using all methods available in the opencv2 library to assess which method is the most accurate.

### Sprint 5

|  |  |  |  |
| --- | --- | --- | --- |
| Sprint Number | Sprint Name | Start Date | Finish Date |
| 5 | Template Matching accuracy Improvement | 11/04/2022 | -- |

|  |  |  |
| --- | --- | --- |
| Task Number | Details | Status |
| 1 | Test template matching using multiple images taken from the game | Completed |
| 2 | Test using threshold values to increase accuracy with template matching method | Completed |
| 3 | Find other methods of increasing accuracy | Incomplete |

For this step the accuracy of template matching was tested. Matches of multiple images were tested. These all proved to inaccurate. Methods provided with OpenCV2’s template matching to raise accuracy such as threshold values were tested to increase accuracy, however these were found to still not accurate enough in most cases.

# Findings & Conclusions

The artificial intelligence proved successful in some ways while being unsuccessful in others. The artificial intelligence was able to find use simple strategies to win battles and even showed some capability of successfully using slightly more advanced strategies. This showed that the decision tree was capable of what it was being used for here and would be able to complete parts of the game.

However, the artificial intelligence was however unable to find attack cards on screen due to the low accuracy of the template matching. This low accuracy meant it often confused cards for other cards and even for other objects on screen.

The decision tree shows a good proof of concept but would have been better served with the use of a more accurate system to read to screen. This would have better shown the ability of the decision tree.

# References

Asquero, 2022. *Advantages and Disadvantages of Artificial Neural Networks | Asquero*. [online] Available at: <https://www.asquero.com/article/advantages-and-disadvantages-of-artificial-neural-networks/> [Accessed 22 April 2022].

Bluestacks, 2021. *What is Bluestacks?* [online] Available at: <https://www.bluestacks.com/about-us.html> [Accessed 9 November 2021].

Britannica, 2022a. *computer science - Algorithms and complexity | Britannica*. [online] Available at: <https://www.britannica.com/science/computer-science/Algorithms-and-complexity> [Accessed 23 February 2022].

Britannica, 2022b. *Pong | electronic game | Britannica*. [online] Available at: <https://www.britannica.com/topic/Pong> [Accessed 4 March 2022].

Brookhaven National Laboratory, 2022. *BNL | History:The First Video Game?* [online] Available at: <https://www.bnl.gov/about/history/firstvideo.php> [Accessed 3 March 2022].

Caubalejo, R., 2021. *Image Processing — Template Matching | by Ralph Caubalejo | Analytics Vidhya | Medium*. [online] Available at: <https://medium.com/analytics-vidhya/image-processing-template-matching-aac0c1cbe2c0> [Accessed 23 March 2022].

European Parliament, 2021. *What is artificial intelligence and how is it used? | News | European Parliament*. [online] Available at: <https://www.europarl.europa.eu/news/en/headlines/society/20200827STO85804/what-is-artificial-intelligence-and-how-is-it-used> [Accessed 18 October 2021].

Hillier, W., 2021. *What Is a Decision Tree and How Is It Used?* [online] Available at: <https://careerfoundry.com/en/blog/data-analytics/what-is-a-decision-tree/> [Accessed 8 November 2021].

IBM, 2022. *What are Neural Networks? | IBM*. [online] Available at: <https://www.ibm.com/cloud/learn/neural-networks> [Accessed 21 February 2022].

Johnson, J., 2020. *4 Types of Artificial Intelligence – BMC Software | Blogs*. [online] Available at: <https://www.bmc.com/blogs/artificial-intelligence-types/> [Accessed 4 March 2022].

Kersley, A., 2021. *What was the Great Video Game Crash of 1983? | BugSplat Blog*. [online] HistoryHit. Available at: <https://www.bugsplat.com/blog/less-serious/great-video-game-crash-1983/> [Accessed 23 March 2022].

LDPlayer, 2021. *LDPlayer - Lightweight & Fast Android Emulator for PC*. [online] Available at: <https://www.ldplayer.net/> [Accessed 17 November 2021].

Mccarthy, J., 2004. *WHAT IS ARTIFICIAL INTELLIGENCE?* [online] Available at: <http://www-formal.stanford.edu/jmc/>.

Microvirt, 2021a. *About Us - MEmu Blog*. [online] Available at: <https://www.memuplay.com/blog/about> [Accessed 24 November 2021].

Microvirt, 2021b. *MEmu - The Best Android Emulator for PC - Free Download*. [online] Available at: <https://www.memuplay.com/> [Accessed 24 November 2021].

Musuem of Play, 2022. *Video Games Archive - The Strong National Museum of Play*. [online] Available at: <https://www.museumofplay.org/video\_games/> [Accessed 23 March 2022].

NoxPlayer, 2021. *Noxplayer – Fastest and Smoothest Android Emulator for PC & Mac – Free and Safe*. [online] Available at: <https://www.bignox.com/> [Accessed 23 November 2021].

Scikit-learn, 2022. *1.10. Decision Trees — scikit-learn 1.0.2 documentation*. [online] Available at: <https://scikit-learn.org/stable/modules/tree.html> [Accessed 4 March 2022].

Stanek, M., 2021. *Understanding AlphaGo. How AI beat us in Go — game of profound… | by Mirek Stanek | Medium*. [online] Available at: <https://medium.com/@froger\_mcs/understanding-alphago-948607845bb1> [Accessed 13 October 2021].

Tamir, M., 2021. *What Is Machine Learning? - I School Online*. [online] Available at: <https://ischoolonline.berkeley.edu/blog/what-is-machine-learning/> [Accessed 12 October 2021].

Turing, A.M., 1950. *COMPUTING MACHINERY AND INTELLIGENCE*. *Computing Machinery and Intelligence. Mind*, .