# Abstract

Ignore for now

# Artificial Intelligence

## Introduction

Artificial Intelligence (AI) is the use of computers to complete various tasks deemed to require “intelligence”, such as writing computer programs, understanding language and identifying images (Nilsson, 1980). There are many applications for AI, from natural language processing to automatically writing computer programs. An example of where AI has been used for natural language processing is Google’s open source language representation model, Bidirectional Encoder Representations from Transformers (BERT).

## Artificial Intelligence in Video Games

AI has traditionally been used in video games to control the decision-making aspects of any non-player character (NPC), “… they can also be used to work in the game’s background, making sure the digital environment runs smoothly, and even on the player’s behalf in roles like tutors or wingmen” (Cass, 2002) . An example of this would be the enemy player in any chess game. In this case, the AI would use a tree search algorithm such as the Monte-Carlo Tree Search to evaluate the best possible move to make. AI has more recently been used to control the player character in video games such as Super Mario Bros, Pinball and Street Fighter 2. These AI programs used a branch of AI called Genetic algorithms and Q-learning to teach the program to play their game.

## Monte-Carlo Tree Search

Guillaume Chaslot et. Al. (2008) describes Monte-Carlo Tree Search (MCTS) as a “best-first search technique which uses stochastic simulations”. They explain that the algorithm simulates a several games in which the player and AI both make random moves; using the information learned from this to infer the best strategy. The algorithm has four main mechanisms to make decide the next move to make: Selection, Expansion, Simulation and Backpropagation. A picture containing screenshot

Description automatically generatedThis is explained further in figure 1 below:

Figure 2.: Monte-Carlo Tree Search Mechanisms (Guillaume Chaslot, 2008)

## Finite State Machines

A finite state machine is used in video games to create a simple AI that can perform a given number tasks. The machine is divided into states and transitions; states being a defined behaviour, such as walking or running and transitions being the conditions that decide what causes a change from one state to another. Applications of finite state machines include controlling the actions of an enemy character or controlling what animations a character is displaying. One proposed issue with finite state machines is the lack of adaptability: “The agents will always make the same moves in the same situations, and the game quickly becomes boring” (Kenneth O. Stanley, 2006).

## Genetic Algorithms

Genetic algorithms can be used in many applications including used to train the application to play the game. The genetic algorithm consistently learns from previous attempts by modifying its parameters based on what has been deemed to work. For example, Jonathan Quirke used a genetic algorithm in this manner to teach a program to play Super Mario Bros. which learned when to jump, when to dodge enemies, etc.

## Q Learning

“Q learning is model-free learning technique that can be used to find the optimal action-selection policy using a Q function” (Raval, 2017). Q-learning is a form of reinforcement learning used in the training of AI’s decision-making policy. The goal of Q learning is to find the optimal function to gain the greatest reward. This function is known as the Quality or Q function. The Q function is constantly updated based upon the results of previous actions. Q learning can be divided into three main functions: observation, action and reward. Observation analyses what it can “see” in the current situation. In the example of a fighting game, observation would look at the current health status, the position and the timer. The action function decides what decision to make based on what it observes. The reward state analyses the outcome from the action and decides how positive or negative that action was.

### Deep Q Learning

### Double Q Learning

# Difficulty Modification

# Street Fighter 2 Gameplay

## Introduction

The game consists of two characters playing against each other in fights. The characters can either be controlled by two players or by a player and a computer. The fight is split into rounds; the winner being the character who wins two rounds first. Each round lasts 99 seconds and the winner is determined by whoever has the most health by the end of the round or whoever depletes the health bar of the other character first.

Figure : Example of fight

## Basic Controls

The game has 12 separate buttons that can be pressed: a, b, x, y, up, down, left, right, l, r, start and select. During a fight, the key bindings are as follows: up, down, left and right control character movement; b, a and l perform light, medium and heavy kicks respectively;

y, x and r perform light medium and heavy punches; start pauses the fight and select does nothing. The punches/kicks can be used on their own to perform a basic attack; they can also be combined with movement keys to perform unique attacks, called special moves, e.g. if the player presses the down key followed by the key facing the enemy and then any punch button while playing as Ryu, the character shoots out a blue fireball. Each character has their own set of special moves that all have unique functions.

## Physics

(character movement, hit boxes, etc.)

# Applied Research

## MNIST Tutorial

|  |  |  |
| --- | --- | --- |
| **Task Number** | **Details** | **Status** |
| 1 | Install Python 3.7.4 | Complete |
| 2 | Install Anaconda 4.7.12 | Complete |
| 3 | Open Anaconda Prompt (Miniconda3) and run the following command “conda install python=3.6” | Complete |
| 4 | Open PyCharm | Complete |
| 5 | Go to the following web address: <https://towardsdatascience.com/image-classification-in-10-minutes-with-mnist-dataset-54c35b77a38d> | Complete |
| 6 | Install Tensorflow plugin from PyCharm | Complete |
| 7 | Download sample dataset | Complete |
| 8 | Install Keras from PyCharm | Complete |
| 9 | Train neural network model using MNIST training data | Complete |
| 10 | Test neural network using MNIST testing data | Complete |

# Ignore for now

Each of these projects had a separate measure of success for the AI; the Super Mario Bros program had to complete a level, the Pinball program had to achieve the highest possible score and the Street Fighter 2 program had to learn the controls for the game and use them to win fights.

Street Fighter 2 Turbo: Hyper Fighting (SF2) was originally released as an arcade game in 1992 but ported to the Super Nintendo Entertainment System (SNES) in 1993.

Delta of health = (damage dealt – damage taken)/maximum health

To keep the game competitive, AI should aim for the delta of health to be as low as possible

# References

Adam Fletcher, J. M., 2018. *Using Python to build an AI to play and win SNES StreetFighter II with machine learning.* [Online]   
Available at: https://www.youtube.com/watch?v=NIG4BZ8VpF4  
[Accessed 15 September 2019].

Cass, S., 2002. Mind Games [computer game AI]. *IEEE Spectrum,* 39(12), pp. 40-44.

Guillaume Chaslot, S. B. I. S. P. S., 2008. *Monte-Carlo Tree Search: A New Framework for Game AI.* s.l.:Universiteit Maastricht / MICC.

Haykin, S., 2009. *Neural Networks and Learning Machines Third Edition.* Hamilton Ontario: Pearson Prentice Hall.

Kenneth O. Stanley, B. D. B. I. K. a. R. M., 2006. *Real-Time Evolution of Neural Networks in the NERO Video Game.* Orlando, Austin: The University of Central Florida, The University of Texas at Austin.

Lippmann, R. P. M., 1988. An Introduction' to Computing with Neural Nets. *ACM SIGARCH Computer Architecture News,* 16(1), pp. 7-25.

Nagle, J., 2018. *Pinball FYP.* Tralee: Institute of Technology Tralee.

Nilsson, N. J., 1980. *Principles of Artificial Intelligence.* 2nd ed. s.l.:Morgan Kaufmann.

Quirke, J., 2018. *Can a Machine Learning Algorithm be Used to Navigate a Level Of Super Mario Bros..* Tralee: Institute of Technology Tralee.

Singh, J. O. X. G. H. L. R. L. S., 2015. *Action-Conditional Video Prediction.* Cornell: Cornell University.

Tarassenko, L., 1998. *Guide to Neural Computing Applications.* s.l.:Neural Computing Applications Forum.

Toutanova, J. D. M.-W. C. K. L. K., 2019. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* 2nd ed. s.l.:Cornell University.