



# Project Title

Project Engineering

Year 4

Tosin Biala

Bachelor of Engineering (Honours) in Software and  
Electronic Engineering

Galway-Mayo Institute of Technology

2020/2021

# Project Graphic (Optional)

## Declaration

This project is presented in partial fulfilment of the Bachelor of Engineering (Honours) requirements in Software and Electronic Engineering at Galway-Mayo Institute of Technology.

This project is my work, except where otherwise accredited. Where others' work has been used or incorporated during this project, this is acknowledged and referenced.

---

## Acknowledgements

Use this section to acknowledge anyone who might have helped during your project if you wish to.

## Table of Contents

|     |                            |                                     |
|-----|----------------------------|-------------------------------------|
| 1   | Summary.....               | 6                                   |
| 2   | Poster.....                | 8                                   |
| 3   | Introduction .....         | 9                                   |
| 4   | Background .....           | 10                                  |
| 5   | Project Architecture ..... | 11                                  |
| 6   | Project Plan.....          | <b>Error! Bookmark not defined.</b> |
| 7   | Heading.....               | <b>Error! Bookmark not defined.</b> |
| 7.1 | Referencing .....          | <b>Error! Bookmark not defined.</b> |
| 7.2 | Notes on Content .....     | <b>Error! Bookmark not defined.</b> |
| 8   | Ethics.....                | <b>Error! Bookmark not defined.</b> |
| 9   | Conclusion .....           | 27                                  |
| 10  | Appendix .....             | 29                                  |
| 11  | References .....           | 30                                  |

## 1 Summary

The project I decided to work on throughout the final semester was a game called Sanio. My project, to its core, was a 2D unity based game that implemented intelligence to the player using reinforcement learning. The objective of the game was to allow the player to finish the game without any external help.

I decided to do a project based on game development was because I was always intrigued by game development. I thought this project would be most enjoyable to complete. I also thought this project was unique compared to my other classmate's projects.

The essential skills I would like to gain when completing my project are to be a competent c# programmer are to gain skills in Unity and get familiar with machine learning to a stage that can create more complex machine learning-based projects.

The approach of my project was to train my player to play the game and go up every level. He would also try to aim to hit the highest score at each level. I then set a schedule for myself. If I finished my project a month earlier, I would implement reinforcement-learning to my players to learn from them and perform different activities to prevent themselves from dying.

Unfortunately, due to time restraints, I was unable to complete the model.

This project is mainly focused on machine learning since I have no prior experience with it. To train my player, I used an open-source tool kit named ML-agents.



## 2 Poster

Poster



### 3 Introduction

This report will reveal the research I have studied throughout the final college year. It will also explain the technologies I've to use to complete my project. My goals for this would be to train my player to finish a game by himself and simultaneously to get a high score each time the player completes a game session.

My next objective is to add levels to the game. During each game level, the player's enemies will become faster and deal more damage. That is an extra feature I decided to implement in my project. Although the model might not be fully complete, I will demonstrate my understanding of the model. This part will be not so easy to train because the learning environment is constantly changing. The Agent will try to find the best policy, thus taking it longer to program and train.

The next objective I would like to implement in my project would be the Ai enemy that trains from the players perspective and prevent them from being defeated. Although this isn't the most crucial feature, I will only try to implement it if I finish my project early.

## 4 Background

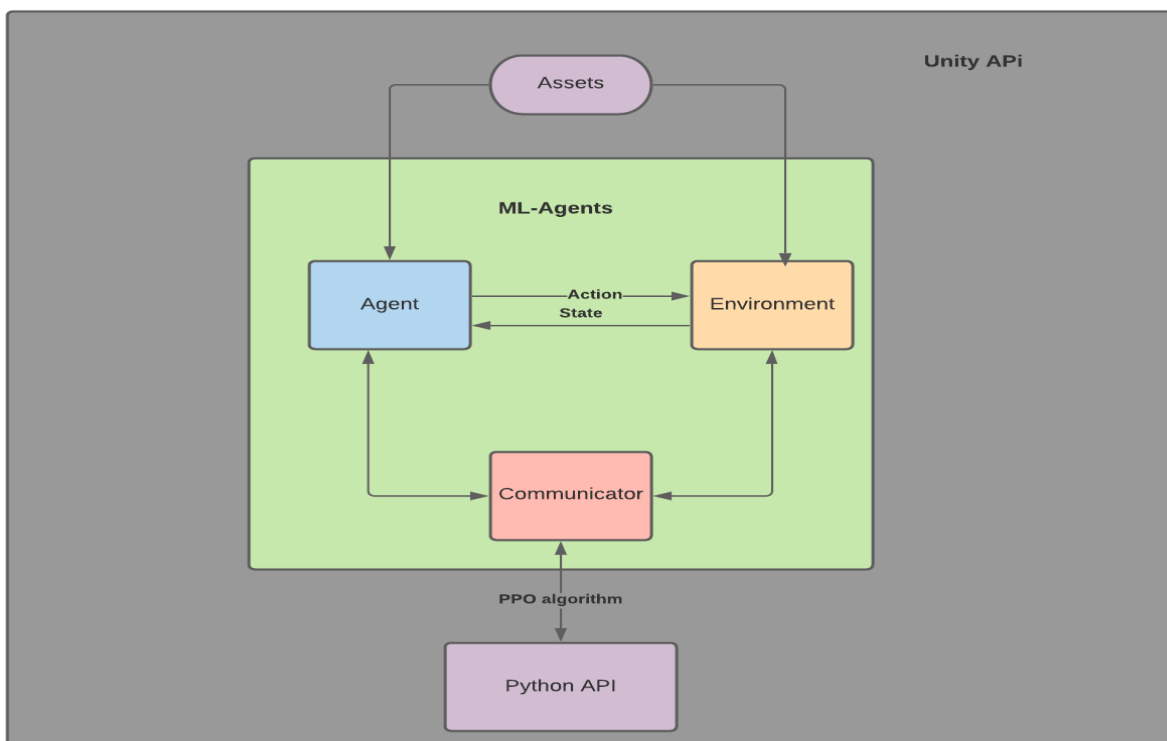
You should change the title of this section to suit your own project subject. The aim of this section is to introduce to the reader any relevant background information that is required for your project.

You may have multiple 'background' sections. Think of any of the questions you had to answer during the research phases of your project – these likely should be addressed in a section like this.

## 5 Project Architecture

Your project architecture diagram should go here. This is an important section, and one most readers of your report will view.

Your diagram should be self-documenting. Use subsequent sections in your report to elaborate on technologies / software / hardware in your diagram.



**Figure 5-1 Architecture Diagram**

## 6 Gamification

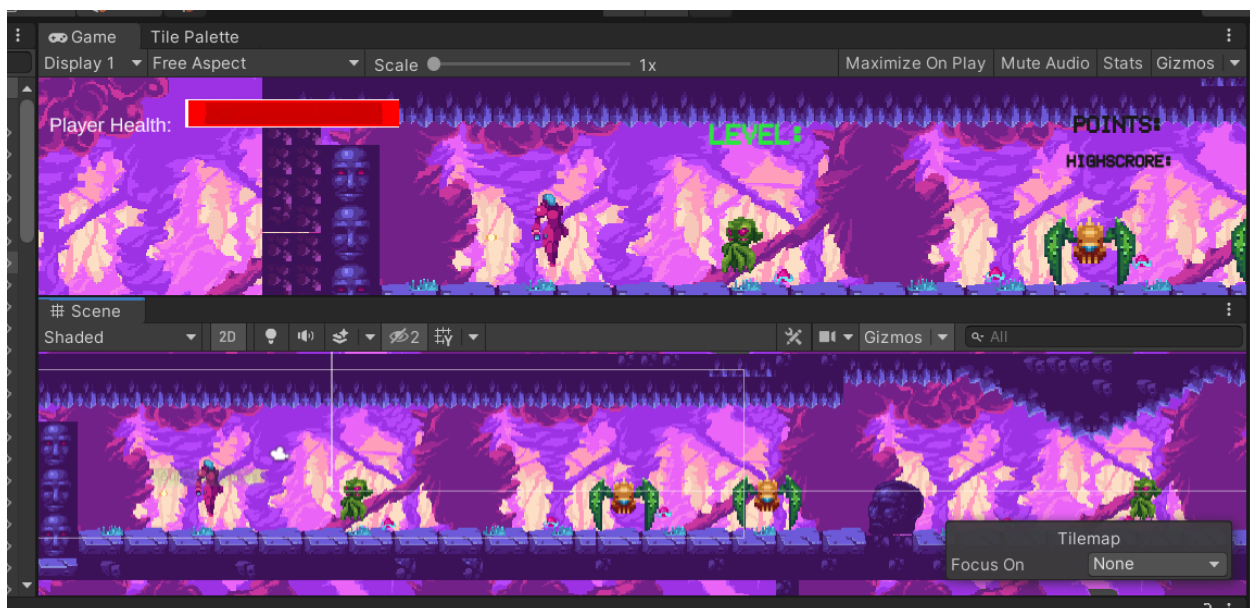
Gamification is the process of applying mechanic or game dynamic in a non-game context. The most prominent non-game context out there is that uses gamification are enterprises.

Gamification techniques are used in enterprises because they engage people in their products, whether by incentives, reward programs, badges, leader board and sometimes free stuff. These rewards motivate people to complete actions.

Another Successful implementation of gamification is Education. Math teachers have been applying some particular way to their classes to make their class environment more enthusiastic. The most popular way they implement gamification would be to add to give a reward at the end of the day based on who got the highest amount of question or create point anarchy based on how the most well-done points.

Trying to apply gamification to my project was difficult because my original idea was to create a 2D shooting game. But back then, early in development, I was not thinking about the gamification techniques; I was more concentrated on the complexity of my project. After some advice from classmates and supervisors, I came to a reionization on how vital gamification is. But as my project was changing throughout the college year, my previous gamification idea did not match my existing game idea at the time.

I decided on a game implantation design that would work for any configuration of my game. The Scoring system is implemented in the game. The player will earn points by destroying an enemy and goal of the game to achieve a high score at every level.



## 7 Unity

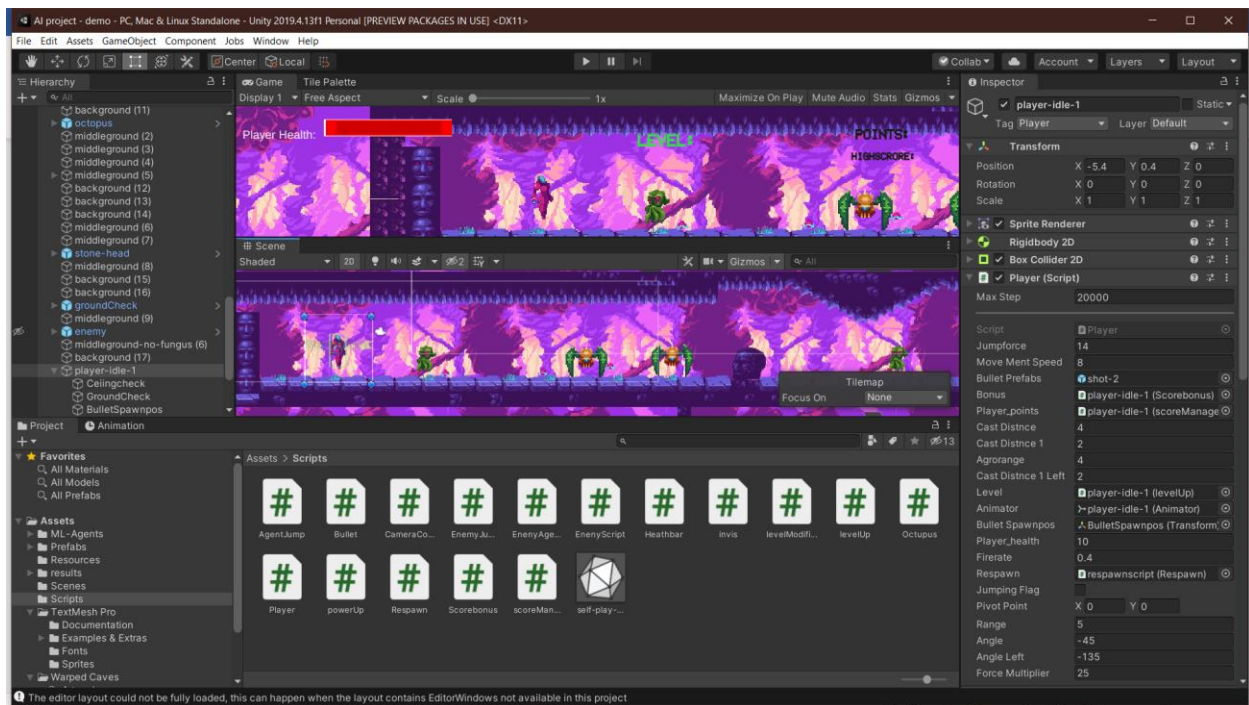
Unity is an American software development company based in San Francisco. Unity is an engine that enables users to design 2D/3D games or app scenes. Unity is not just for game design. It is a use or simulation design for businesses. Unity allows users to interact with assets with code and imported animation. The programming language that I needed to learn to use Unity was C#.

Another game engine I was planning of using was the Unreal engine. The unreal engine was a more powerful game engine than Unity and programmed in a language I had experience in, which was C++. I decided to go with Unity because I found out Unity is more user-friendly than the Unreal engine after some research. Also, because I was new to game development, I would want memory crashes due to insufficient memory management to disturb my progress during the project if I used the Unreal engine.

Instead of using language-based libraries that made it possible to create games, e.g. PyGame for Python, game engines are much faster efficient when creating games. One major factor that differs from those two would be Unity has a built-in physics engine, making it easier to program complicated game projects.

An example of a method that access physics in my project is Rigidbody. It allows that game object assigned to it to give the game object access to gravity forces mass and Velocity. I also implanted some other physics such as colliders to check if a game object has collected with it and Raycast that shoots out a ray and from the action if I hit the specific thing.

In Unity, there are three fields Inspector, Hierarchy and Scene. The Hierarchy displays all the game object that is in your Scene. The Scene views show what the user can see when they start the game. The inspector shows the properties of a specific game object.



## Vectors

In-game development, a vector is used to meshes calculate distance. A vector represents two points from origin to and 2D plane. The line that represents the space from the two points is called the magnitude. Unity can create vectors at any dimension. Since my project is a 2D game, I was mainly using 2D vectors. I use vectors in my project to get the position and my player. I also use it for distance calculation and to add force to my game object.



## Git

Git is a version control system. A version control system allows users to save the code that we are working on. Git uses features to revert subtle changes to their previous state or revert the entire project and compare changes over time. Git also has a part if you lose files or can quickly recover them. In git, there are three types of version control local version control, centralized version control and distributed version control.

Local version control is stored locally in a database and is the most common type of version control. An individual usually uses this type of version control. For my project, this is the method I used for my project

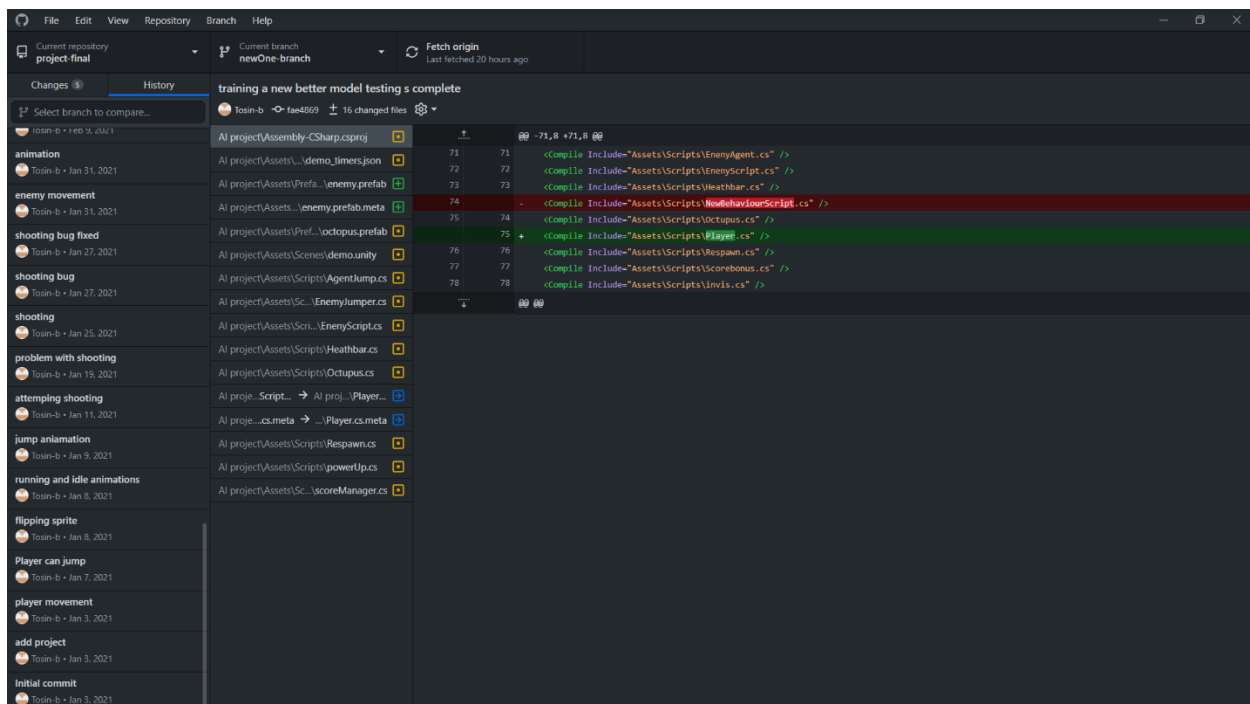
Centralized version control a version-controlled that you would usually use in a team. Code is then managed in the central VSC server. This type of control system gives the administrator more control of the system.

Commands I used during my project were git add, git init , git status, git commit. Git add, adds all all untracked and new files and preparing for the next stage to commit. Git commit commits your project to a repository. Git init,to create a new repository. Git status gets the status of a file.

## GitHub

GitHub is a platform that helps programmers work together. GitHub has a massive amount of features, but because my project was an individual project, I was limited to the number of features I could use. GitHub allows programmers to share their projects that get stored in a repository to the world. One issue I've come across during my project was a merge conflict. Merge conflicts generate when the previous code has not pushed the local repository and attempted to commit more existing code. This issue was complex, too, because when starting my project, I was working on the master branch with I found out later this was not the most efficient way to use GitHub. I solved this problem was to create a new branch and clone my previous repository, save my clones repository into a new one.

Another problem was I had no gitignores in my repository, meaning Github read every single file in my unity project, making the size of the project bigger. So big that it surpassed the max size you can send, which was 100MB.



## Machine learning

Machine learning is algorithms that give computers the ability to learn from existing data.

Machine learning is significant in the world we live in and constantly evolving. We use machine learning from self-driving cars to Amazon, Netflix recommendations, to fraud detection. The learning system is broken into three steps decision process, error function, a model optimizer process.

The decision process finds a pattern in the data then attempts to produce predictions.

The error function is responsible for comparison to validate if the generated model is accurate is fine.

The model Optimization process optimizes the model to fit data points in a training set by adjusting the weights. The algorithm repeats until the threshold has been met.

### Machine learning Methods

There are different types of machine learning algorithm and methods in my projects, but the kind of method I decided to learn was reinforcement learning.

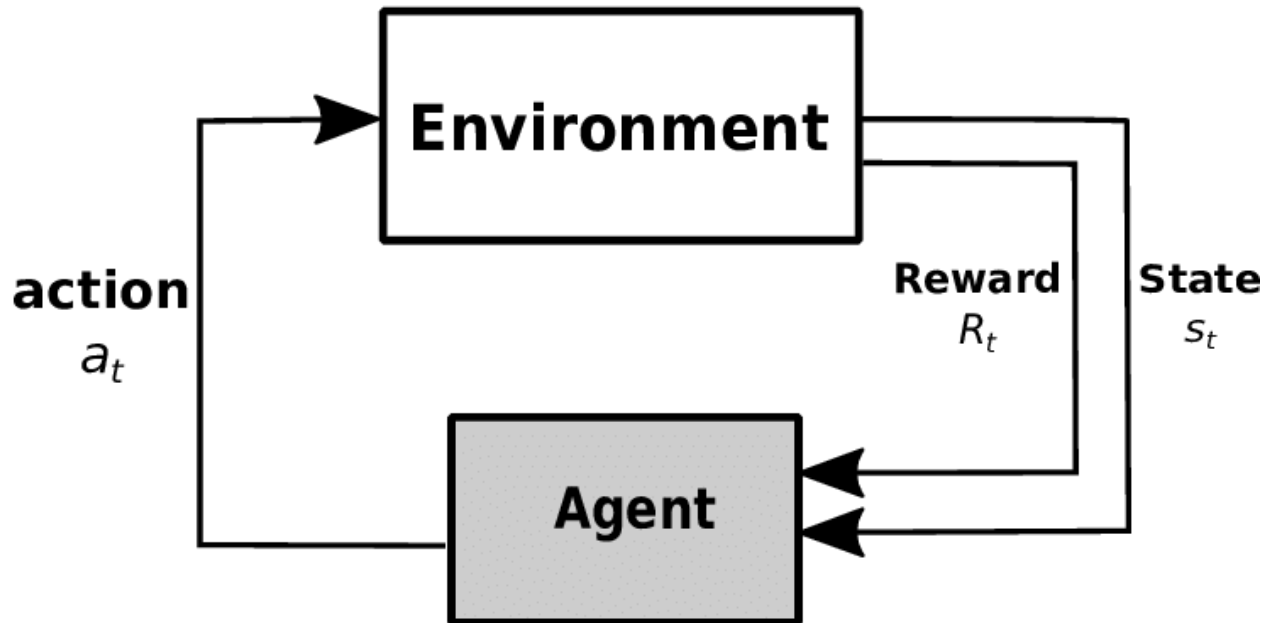
Reinforcement learning is the process of an agent receiving a reward based on the action it took. That reward is saved, and the Agent then tries to earn as many rewards as possible by taking action. The goal of reinforcement learning is to increase your cultivated reward steadily. Thus your Agent gets smarter. I decided to use this machine learning method because I wanted my player to learn from the environment and past experiences with no Prior knowledge.

In reinforcement learning, there is an environment. An environment is where the Agent is. The environment is then given a state to action.

A state is when an environment provides a situation in the environment to the Agent.

The Agent then determines an action to take based on the state that was given to the Agent.

During each state, a policy is generated. The policy calculates the best action to take that will earn you the most amount of rewards. The longer you train an agent, the better your policy is by calculating the best cumulative reward for future rewards



Q function can find the optimal action selection policy. It takes any state-action pair and gives you a real value and find the best action.

There are two other types of Reinforcement learning Supervised learning, unsupervised learning.

Supervised Learning is the when a computer learns how to map an input to an output based on the input output pairs.

Unsupervised learning learns by observing patterns from untagged data.

## Neural network

The human brain inspires neural networks. Neural networks can take in data train themselves by identifying a pattern and then predicting the outputs. A neural network is made up of neurons. The neurons job is to compute the weighted average of its input.

A neural network has three layers input layer, an output layer, and a hidden layer. When observation is fed into the neural network, they go through the input layer first. The neurons are then connected through channels. Each channel is given a weight. In the input layer, the inputs are multiplied to each neuron corresponding weight. The sum is then set to the hidden layer. These individual sums are called a bias. Then the bias value is then sent to an activation function. The role of an activation function is to determine a neuron is activated or not. An activated neuron transmits data to the next neuron layer through channels. This process is called forward propagation. At the output layer, the neurons with the highest probability will be the predicted ones.

Backpropagation is when the neural network compared the predicted result to the input and generates the error magnitude. Then transfer backed into the neural network.

For my project, I had to tune my hyperparameters to change the neural network. Num epoch hidden units num layers

Num epoch is how many times the neral network runs to generate a typical prediction range from three to ten. I implemented. The bigger the number, the longer it will take to train.

The hidden unit determines how many units are in each neuron layer typical range was from 32 to 512. I chose sixty-four. I tried it with thirty -two and my Agent didn't seem to learn very well. This was because the about of observation I was feeding into the neural network was not efficient at 32.

Num layers are how many hidden layers are represented in yr neral network I decided to implement three. From using trial and error, I found three to the be most efficient one based on my Agent.

## Hyperparameters

Hyperparameters interpret data. ML agents generate you a default one but the more observation you are feeding into your neural network, the poorer your neural network will perform. I will explain below the hyperparameter tuning I've implemented into my project.

**Learning\_rate** speeds up the neural network update weights depending on the number you insert. Typical range is  $1e-5$  to  $1e-3$ . I chose my learning rate to be  $1e-3$  because I wanted to update many times during training, thus slowly improves the Agent. I have a maximum rate of 0.3 which will result in the loss of previous past experiences.

**Learning\_rate\_schedule** determines your environment. If your environment does not change it should be linear but if your environment is constantly changing, it should be constant. I previously set this parameter to linear. Later on, I found out that I should have set it to constant because my environment was changing.

The **batch size** determines how many of the agents' experiences are fed into the neural network. To produce a neural network update. I set it to 64.

**Buffer\_size** determines the policy update

**Beta** determines how random the policy is at the beginning of training. if this value is high, the agent explore more of the environment

Mlagents

Mlagents is a open source toolkit that I used to train my Agent. It provide simple to use python api to train your Agent. The python api contains machine learning algorithms. The algorotriths then look up how.

In Mlagents your are given three fucntion onActionrecieved, CollectObservations, onEpisodeBegin, hueristic.

OnEpisodeBegin is when a a episdoe end it calls this functiona nd depeinding on what you wnt your program to do you would call it in this fucntion. Usually you would just reset the scene but i had to implement to reset the score and to bring you back to level 1.

OnActionRecieved tells the Agent what he is able to contorl. In this fucntion I wuld add rewards depending on the action that was taken. Since I want to my player to walk back and forward I added forces to the players x and y axis so it can jump.

Collects observations colleted the observations that I was going to use to feed into my neural netowrk. These were the play x and y position so the player could learn to move and jump. Raycast to check if a an enemy is near the player. the players health and points to allow the Agent to keep aware if the health.

With the hueristic feature I was able to test my enivorment for bugs before I went to train ym Agent. I would do this by taking control of the jumping and the movemnt that the Agent has control.



ppo

Virtual environments

## 8 Conclusion

Write a short conclusion. What is the outcome of the project? Perhaps you have a product prototype, or some results, or a demonstratable system.

Do not use your conclusion to tell the reader what you might have done if you had more time, but keep it focussed on what you actually have done. You can mention future opportunities for further development of the work, but keep this part short.

## 9 Appendix

## 10 References

- [1] H. Kinsley, "Reinforcement Learning," PythonProgramming, [Online]. Available: <https://pythonprogramming.net/q-learning-reinforcement-learning-python-tutorial/>. [Accessed 02 02 2021].
- [2] [Online].
- [3] MakeSigns, "Scientific Posters Tutorial," [Online]. Available: <https://www.makesigns.com/tutorials/scientific-poster-parts.aspx>. [Accessed 09 02 2021].
- [4] Arduino. [Online]. Available: <https://www.arduino.cc/>. [Accessed 09 02 2021].
- [x] J. C. Russ and F. Brent Neal. *The Image Processing Handbook*. 7<sup>th</sup> ed. Boca Raton FL: CRC Press, 2017.
- [x] S. Lippman, J. Lajoie and B.E. Moo. "Classes" in *C++ Primer*. 5<sup>th</sup> ed. Massachusetts: Addison Wesley, 2013.
- [x] IEEE Signal Processing Society. "Signal Processing for 5G", *YouTube*, 2019. [Online]. Available: [https://www.youtube.com/watch?v=uca6X4Ykcmg&list=PLcZOnmyglalacL9YqkhyufLQGIW\\_C78Os&index=6](https://www.youtube.com/watch?v=uca6X4Ykcmg&list=PLcZOnmyglalacL9YqkhyufLQGIW_C78Os&index=6). Accessed: Feb 2, 2021.
- [x] Digilent. "Basys 3 Reference Manual", *Digilent Reference*. [Online]. Available: <https://reference.digilentinc.com/basys3/refmanual>. Accessed: Feb 2, 2021.
- [x] P. J. Ashenden. *Digital Design (Verilog): An Embedded Systems Approach Using Verilog*. Burlington: Morgan Kaufmann, 2007.

[x] M. Lynch. "Discrete Fourier Transform (DFT)", Lecture, Digital Signal Processing, Galway-Mayo Institute of Technology, Galway, 2020.

[x] MRC Centre for Global infectious Disease Analysis. (2021). *Covid-Sim*. [Source Code].

Available: <https://github.com/mrc-ide/covid-sim>. Accessed: Feb 2, 2021.

[x] OpenCV. "Face Detection Using Haar Cascades", *OpenCV-Python Tutorials*. [Online].

Available: [https://opencv-python-](https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_objdetect/py_face_detection/py_face_detection.html#face-detection)

[tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_objdetect/py\\_face\\_detection/py\\_face\\_detection.html#face-detection](https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_objdetect/py_face_detection/py_face_detection.html#face-detection). Accessed: Feb 2, 2021.



















