

**Computer Games Development**

**SRS**

**Year IV**

## 

**Jack Fennell**

**C00220386**

### **Contents**

[**Contents**](#_k5ra82gw51bx) **2**

[**Acknowledgements**](#_px0t4egnugae) **3**

[**Project Abstract**](#_1ca5bua76g6l) **3**

[**Project Introduction**](#_l1awhyzgwxzq) **3**

[**Background**](#_ft3tf8e9i00) **3**

[**Project Description**](#_tdzi1s4oelii) **3**

[**Define the Application**](#_u5qweyx9vr46) **3**

[**What is the application supposed to do**](#_9mqvs7ecrszr) **3**

[**Who is going to be using this application**](#_oh8ramtvx8r7) **4**

[**Context Diagram and Use Cases**](#_dvpsxb7nrb2n) **4**

[**Metrics**](#_gg4owe4itdnp) **4**

[**Is there a precedent for this application**](#_ly6ngae8w0ku) **4**

[**Project Milestones**](#_g875aybv203y) **4**

[**Project Review and Conclusions**](#_muq4t1my9p0o) **4**

[**References**](#_gyjjnvuraqt) **5**

### 

### 

### 

### 

### 

### 

### **Acknowledgements**

I would like to thank the following people who assisted in completing this project including:

Dr. Oisin Cawley who assisted and helped a lot during this project through resources and assistance along the whole process of the project.

### **Project Abstract**

This project will be focusing on the problem domain of artificial intelligence in particular the use of artificial neural networks (ANN). ANNs structure and operation is often represented as a brain of a human, as it is made up of neurons or connections that send signals to each. ANN’s are used in a huge variety of fields such as data processing,image recognition and speech recognition. My focus will be on the use of an ANN in a video game to play the game using the AI. The ANN will need training through algorithms in order to deliver the best results for the AI.

The goal of this project is to understand how ANN’s work in a game and how different algorithms affect the ANN’s operation and training.

### **Project Introduction**

My project is to create and train an artificial neural network to play Geometry Dash with both a Genetic and a Backpropagation algorithm training the network and comparing their results. Geometry Dash is a very basic endless runner game which requires the player to jump over obstacles to proceed. I will implement the game with Python to work with Tensorflow and TFLearn as additional libraries to compute the necessary data for the neural network. I chose this project as I wanted to see and use artificial neural networks for my project as they represent the connections made in a brain and recreate and control entities based on the neural network’s brain. A neural network must be trained for all different possible outcomes so repetition is vital for it to learn the correct path. Neural networks were once regarded as the best method to train AI but this has changed with new options entering the field and they don't see much use in gaming. Neural networks are now used in numerous fields due to solving numerous problems efficiently, such as medicine, voice and image recognition and some banking and financial uses.



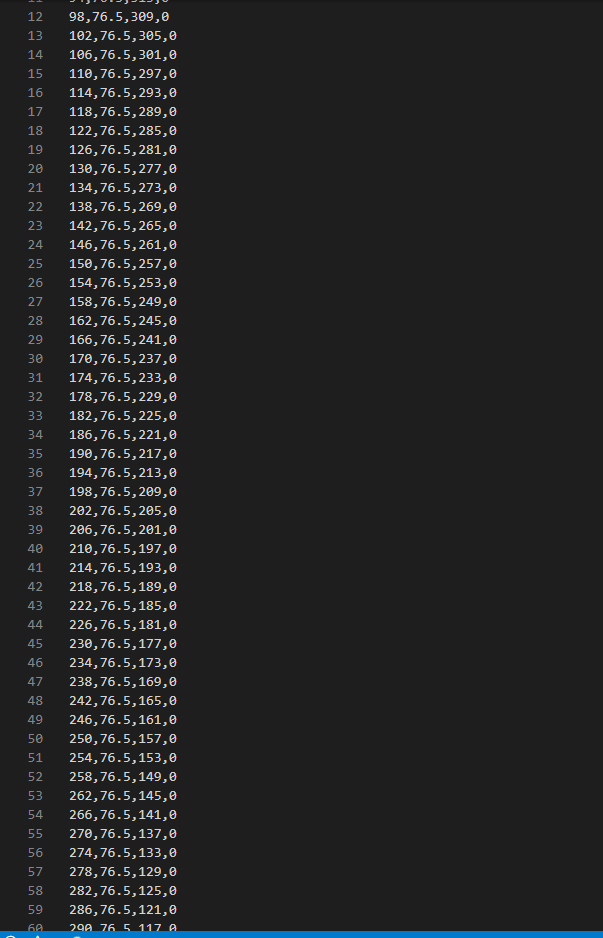
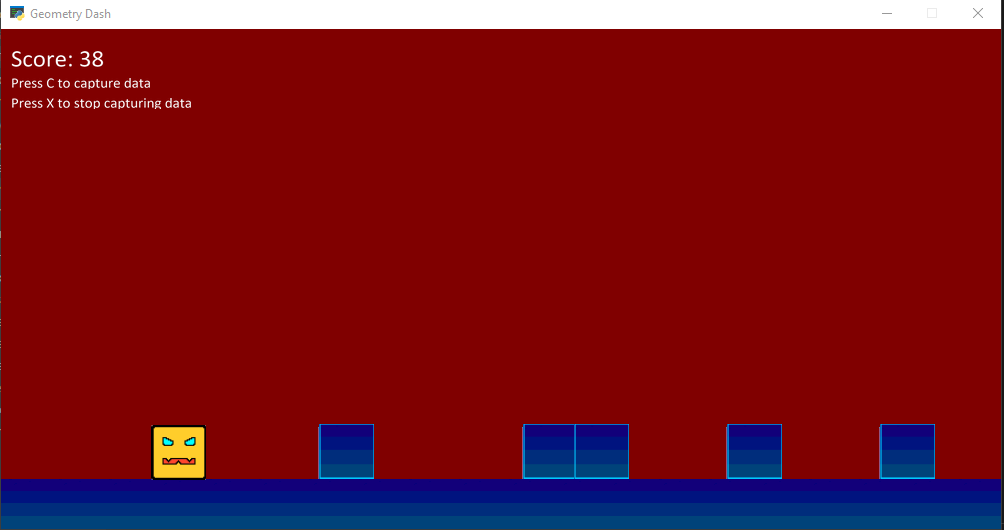
Fig.1 Example of Geometry Dash.

**Background**

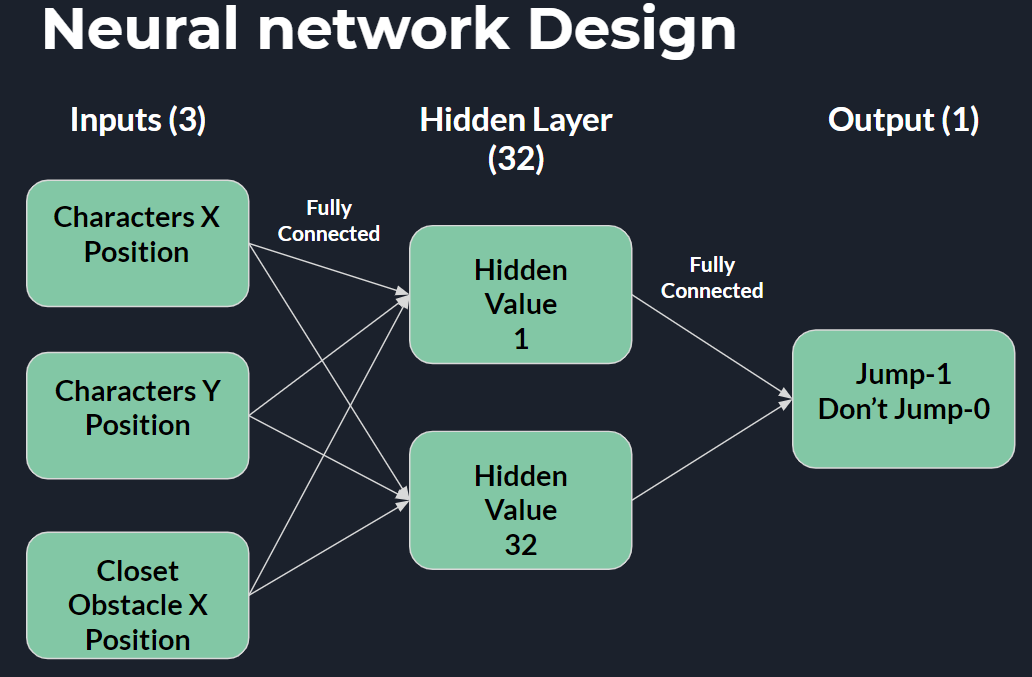
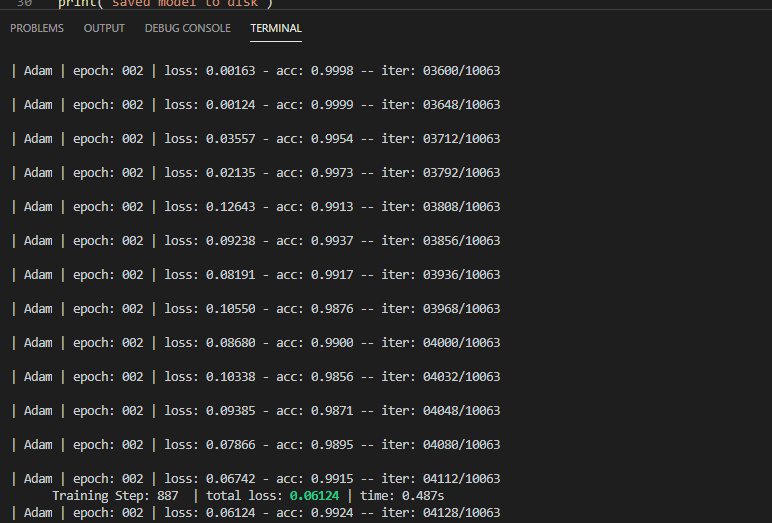
Video games have a lot of different methods available to create the AI for the game. For example finite state machines(FSM) could be one such option. FSM breaks down the AI’s behaviour into states and transitions from one state to another. This can produce the desired behaviour of an AI but after multiple interactions with a FSM AI it can become predictable. Another option could be a rule based system(RBS). RBS followed predefined rules,which execute on certain events in the game making it easy for the AI to work but again it becomes predictable like the FSM after multiple interactions. Both these methods don’t allow for dramatic change in the game environment. This is where ANNs come in. ANNs copy the natural process and create a sense of realism to the AI. ANNs require a lot of data inputted to train but with this can create complex behaviour while also learning as the game is played. The only problem with ANN is that they require a lot more processing power to create this complex behaviour compared to both the FSM and the RBS.

**Project Description**

My finished version of Geometry Dash had a player block which auto moved right as the game began. Obstacles would then spawn in at random positions to block the player's path. When the player collided with the side of these obstacles they would go to the game over screen and then attempt the game again. The player can jump over these obstacles or land on them to proceed while gaining a score. While the game is running the player can hit the C key to begin to capture data while the game is played that is appended onto a csv file. The data collected is the player's current X position,current Y position, the closest obstacle’s X position and whether the player jumped or not(Indicated with a 1 for jump and 0 if not).



I then took this data and fed it into an ANN I created in TFLearn. The ANN I created took 3 inputs(current X position,current Y position, the closest obstacle’s X position). The hidden layer had 32 nodes whose values(weights) would be calculated by running through the inputs and delivering certain output either 1 or 0 based on the jump data collected. The ANN took the data collected and went through all the data numerous times based on the epoch set. The final accuracy of the network was usually about 0.99 with a loss of about 0.06124.



The original project I began to develop was to create a ANN to train a swarm of bees but as I progressed through it I saw the project getting a bit complex and difficult as I couldn’t grasp the understanding of the ABC algorithm so I narrowed down my project to focus on the merits of an ANN in the use of a simple video game.

My game was going to be developed in C++ but as I explored options for creating the neural network I decided to use Tensorflow and TFLearn which worked best through Python and had the most documentation for support and help.

My finished project lacked complexity in the Geometry Dash game when it came to obstacles as there were no spikes or higher placed blocks. It also didn't have the algorithms playing the game without the player control.

I learned a lot about Python code in the process of this project as before this project my use of Python was small and for very basic programs. This learning was assisted by the use of the Arcade library which helped me develop the game of this project with a lot of ease.

**Overview**

**Define the Application**

**What is the application supposed to do**

**Who is going to be using this application**

**Context Diagram and Use Cases**

**Metrics**

**Is there a precedent for this application**

**Project Milestones**

Sprint 1 (14th October - 28th October): Begun research into the project, began research document.

Sprint 2 (18th November - 2nd December): Began working on the initial code created the base Geometry Dash game with basic physics.

Code Demo and Presentation (9th December):

Sprint 3 (6th January - 24th February):

Sprint 4 (9th March - 28th April):

Code Demo and Presentation 2 ( 1st April ):

Final Presentation and Code Demo(30th April and 1st May):

**Project Review and Conclusions**

The game development went well in regards to the Geometry Dash game. The only reason it lacked some complexity was due to the fact that training for the neural network did not get past the first iteration of implementing it to play the game in its basic form only jumping over the basic obstacles. The neural network came together quite well with the help of TFLearn, although the one thing I would have corrected with it is the amount of hidden nodes in the hidden layer as 32 was a very excessive amount for the basic neural network that I had created. The project lacked implementation of any AI playing the game with either the BP or GA method so due to that I could not compare any algorithms for the main purpose of the project. If I was to approach this project again I would have worked harder, not procrastinated and sat on my hands when work needed to be done. I also would have focused more on fully understanding the area of work as it took me quite a while to grasp some of this project area.

**References**

|  |  |  |
| --- | --- | --- |
| **Referenced Publication** | **Citation** | **Reference** |
| Report | (Simon D. Mikulcik  2016) | Simon D. Mikulcik  Application of Neural Networks for Intelligent Video Game Character Artificial Intelligences |
| Report | (J Mänttäri and  J Larsson.  October 2011) | J Mänttäri and J Larsson.  Applications of Artificial Neural Networks in Games;  An Overview. |
| Website | (Collection of Authors) | Multiple Authors 2005. Artificial Bee Colony (ABC) Algorithm. (URL https://abc.erciyes.edu.tr/index.htm)  (Accessed 23 October 2019). |
| Report | (Zhamri Che Ani, Azman Yasin,  Mohd Zabidin Husin and Zauridah Abdul Hamid  2010) | Zhamri Che Ani, Azman Yasin, Mohd Zabidin Husin  and Zauridah Abdul Hamid.  A Method for Group Formation Using Genetic  Algorithm |
| Report | ([Partha Pratim Sarangi](https://www.researchgate.net/profile/Partha_Sarangi), [Abhimanyu Sahu](https://www.researchgate.net/profile/Abhimanyu_Sahu3) and [Madhumita Panda](https://www.researchgate.net/profile/Madhumita_Panda). 2014) | [Partha Pratim Sarangi](https://www.researchgate.net/profile/Partha_Sarangi), [Abhimanyu Sahu](https://www.researchgate.net/profile/Abhimanyu_Sahu3) and [Madhumita Panda](https://www.researchgate.net/profile/Madhumita_Panda).  Training a Feed-Forward Neural Network Using Artificial Bee Colony with Back-Propagation Algorithm |
| Website | (Garych 2020) | Garych  How to Compare Machine Learning Algorithms. |

### 