**A picture containing logo

Description automatically generated**

**Soft Computing Project**

**Topic**

**Space Shooter Game Bot using Neural Network**

**Submitted to-**

**Dr. Sukhvir Kaur**

**Made by-**

**Gaurav Pandey**

**11909570**

**RKM047B65**

**INTRODUCTION**

**This is the project on Soft Computing subject. The name of the project is “Space Shooter Game bot using Neural Network.” The language used to make this project is PYTHON. I made the game using pygame library and the neural network was designed from scratch.**

**The libraries used in this project are, pygame, NumPy, random and math. Pygame is used for creating the game environment along with the random module and the math module. NumPy module is used in the implementation of the Neural Network.**

**The Neural Network which I used is the Back Propagation Neural Network (BPNN). The method of learning which this Neural Network use is Supervised Learning.**

**This project is an implementation of the neural network in the sector of video games. This report consists of the full explanation of the code along with the snapshot of the code and the output.**

**This report is divided into three major segments. First part is mainly about the game development. Second part is all about the Neural Network. Third portion is all about the explanation of the Neural network which is used in this project.**

**GAME DEVELOPMENT**

**This project is all about the game development and the implementation of Neural Network in automating the game. The game developed in this project is the Space Shooter. I created the game using PYGAME library. The project is completely made in PYTHON. Below is the glimpse of the game.**

**A screenshot of a computer

Description automatically generated with medium confidence**

**Now we will discuss the steps I followed in making this game.**

**In building a game, at first, we need to build a game screen. At first, I imported all the necessary libraries that is required in building the game. For example, pygame, random, math and NumPy. Once I imported all the libraries in my python file, I created my game screen using the inbuild command in the pygame library.**

**The game screen is ready, and now we need to edit the game window. We can edit the background, Title of the window and the icon on the game window. I set the game background using an image file. I changed the game window title to “Space Shooter”. I changed the icon of the window using an image file.**

**I set the game window size to 800x600 pixels, while creating the game window. Now it’s the time to create our players and enemies of our game. I downloaded some game sprites for my player and my enemy. I set the default entries of the player and the enemies and loaded the image of the enemies and the player in the python file.**

**Now I created the game loop, which is necessary for continuous execution of the game. The problem with the infinite loop is that we can’t close the game window and our system enters in a deadlock state. In order to tackle this situation, I created a game event for quitting the game.**

**Now since we have a stable game window, we are ready to set our game player and the enemy to our game. I loaded the player image inside the game loop and along with that I managed the x and y coordinates of the player.**

**Now it’s time to create the game enemy. I created three enemy for my game. It means there will be three enemy rendered in the game screen. I set up the x and y coordinates of the enemy and created a loop to create a new enemy in the game when an enemy passes the game screen, or it is killed by the player.**

**Now we need to build the bullets for our player. I used an image file to create a bullet for our player. I set the x and y coordinates of the bullet from where it is supposed to be fired. I set the condition that if a bullet hits an enemy or it crosses the game screen it will vanish, and a new bullet can be fired.**

**Now we need to make our enemy mortal. I setup the conditions for the death of the enemy, and each time our enemy dies our score will be increased by 1. All the time our enemy escapes our player bullet and leaves the game screen the score value is decreased by 1.**

**Now I wrote the code using the pygame to display the score value on the game screen. Now I set the game music using mixer in pygame.**

**Now we are done with the game development part and now I moved to the Neural Network portion.**

**NEURAL NETWORK**

**I used the Neural Network to make the bot for our game. Th neural network which I used to create the game bot is the Back Propagation Neural Network (BPNN). Now we will look on the Back propagation neural network first and them we will look and the implementation portion.**

**BACK PROPOGATION NEURAL NETWORK:**

**Backpropagation is the essence of neural network training. It is the method of fine-tuning the weights of a neural network based on the error rate obtained in the previous epoch (i.e., iteration). Proper tuning of the weights allows you to reduce error rates and make the model reliable by increasing its generalization.**

**Backpropagation in neural network is a short form for “backward propagation of errors.” It is a standard method of training artificial neural networks. This method helps calculate the gradient of a loss function with respect to all the weights in the network.**

**Graphical user interface, website

Description automatically generated**

**Now we will look on the different terms we need to learn before we move forward in this topic.**

* **Simplifies the network structure by elements weighted links that have the least effect on the trained network**
* **You need to study a group of input and activation values to develop the relationship between the input and hidden unit layers.**
* **It helps to assess the impact that a given input variable has on a network output. The knowledge gained from this analysis should be represented in rules.**
* **Backpropagation is especially useful for deep neural networks working on error-prone projects, such as image or speech recognition.**
* **Backpropagation takes advantage of the chain and power rules allows backpropagation to function with any number of outputs.**

**The neural network which I implemented for the project has two input layer three hidden layer and one output layer. Below is the glimpse of the code that I wrote to implement the Neural Network in my program.**

**A screenshot of a computer

Description automatically generated with medium confidence**

**The concept behind the implementation of the neural network is that there will be three enemies rendered on the game screen and the player need to shoot them identifying the closest enemy to the ground. I pass three inputs and assigned random weights to the links between the input and hidden layer and the hidden and the output layer.**

**I used the sigmoid functions as it is used in the BPNN. I performed 5 lakh iterations to train my model and get a good accuracy.**

**I used those results to create the game bot. This is the complete explanation of my project.**

**CONCLUSION**

**This project was a complete built from scratch and the content inside it are unique. Building this project help me to learn several new concepts in python and different libraries. I came across many challenges while building this project.**

**I would like to thank my professor Dr. Sukhvir Kaur, for giving us this opportunity to enhance our knowledge. I would also like to thank my family members to keep me motivating during this journey.**