Denzill Loe

SFSU ID: 920606619

GitHub username: Denzill7

CSC 317-04

12/04/2020

JavaScript Game Assignment

The game I made for this project consists of a simple one-screen platformer shooter. Everything except the bullets and the text for the score is drawn using a tile sheet that I modified in Photoshop from a game tutorial I found. Initially, I had coded a project with multiple JavaScript files that had separated classes for the controller, model, and view, but I eventually scrapped that and wrote JavaScript code that was entirely contained within my html page.

I reused the logic I wrote for the controller in my previous project and inserted it into this project I made now. The controller takes in four buttons: the z key, the x key, and the left and right arrow keys. Z is used to jump, while x is used to shoot the player's gun. Left and right are used to move the character left and right respectively.

The rules of the game are very simple: shoot and avoid the enemies. There are two types of enemies I made. The first one is the fox. The fox is always grounded and will pick a random spot in the world to walk towards. After 10 seconds, a second fox will spawn on the right side of the map. Shooting the first fox will make it disappear and respawn on the left side of the map, and shooting the second fox does the same on the right side. Shooting a fox earns you 50 points. The second enemy is the ghost. The ghost hovers far above the player and will swoop down if the ghost is directly above the player. The only way to defeat the ghost is to lure it down to the ground and shoot it while it rises back up. Shooting the ghost will make it disappear and respawn on the left-side of the map. Shooting the ghost earns you 100 points. If the player touches the ghost or either of the foxes, the game ends and endgame text with your final score is displayed.

The thing I struggled the most with was with the collision detection between the bullets and the enemies. It's still not perfect, and sometimes a shot can go through a character, but it reliably works now. A collision is calculated by using the coordinates of the bullet and comparing it to the coordinates of the sprite of my game object. If a collision is calculated, both the enemy and the bullet are transported outside of the map. The bullet will automatically be destroyed once it reaches beyond the boundary of the map, but the enemies will stay moving outside of the map space until they respawn back inside the map.

I have also included a short video of my screen capture showing me playing the game. I hope you enjoy it.