Jason Mayotte

IGME.330.01

Project 2 Documentation

3/26/2018

Project 2 Documentation

Our original idea for this project was to make a game similar to that of snake. Within guidance we were steered into doing a project on centipede instead. I was scared of that idea at first, however as we worked on this project I think that I started to enjoy working with centipede more than I would have with snake.

For media requirements, I feel as though our projects meets most of the requirements given in this particular category. Our project did not use a sprite sheet or particle system. Although not much, we did apply a box shadow to the items in the canvas. This namely helped the player’s avatar stand out against the rest of the screen. Our movement, our animations, were smoothed by using delta time. We also did not use “window.cancelAnimationFrame()” to any effect in our code. The graphics were all made by me on website/software called Piskel.

In regards to our interaction requirements, I feel that we met the prerequisites given to us. The player can control the ship sprite using the arrow keys and can shoot using the spacebar. There is nothing complex about our controls. Our controls do run quite smoothly thanks to the use of key daemons and its ability to capture multiple key inputs.

Our project gives the user very clear instructions on how to play the game at the title screen. I think that our game is intuitive enough to understand that the player has to find a way to not get hit by the centipedes coming towards them. Our game is not too difficult, however the game will ramp in difficulty the more centipedes are eliminated. Each time an entire centipede is wiped out, another larger centipede will spawn at the top of the screen along with more mushrooms to block your shots. The game keeps track of your score in the top left corner in order to keep the player in the know of how they are doing. Our game has a total of four screens: title, main game, pause, and a game over screen. Our name will be found on the title screen of our game.

Regarding the mechanics of the game itself, we need plan for majority of them. What was even more surprising is that we did actually manage to meet a lot of our goals regarding the gameplay of centipede. Our aim of the project was to make a recreation of the arcade classic centipede, and to a fairly large degree, I believe we accomplished this goal. Since our project is an actual game, the user and their ability to play the game is indeed in direct control of their experience with the game. The aesthetics are fairly consistent overall. The only thing that differs from the more 8bit of chiptune style of the project is the background of the main game. The only reason for this is that creating a pixel background of that size would be something of a magnitude that I have never accomplished before. I do think that what I created still works for the most part as it is not the primary focus of the main screen. From personal experience of playing the game, I found myself feeling panicked the more and more the centipede came closer, similar to that of the arcade game. But besides that, I found myself having fun with the game, and I hope that anyone else who plays the game shares those feelings. Since our game is simple in concept, I believe that it is approachable by anyone. The player’s choices revolve around split seconds decisions in order to stay alive. The more the user plays the game, the more they can look and see the pattern in how the centipede moves. By doing this the player can learn how to predict the movements of the centipede and plan accordingly. Our sprites, primarily our centipedes, have a unique aspect to them as they will always change their image on a given collision. These sprites bounce off other sprites and walls and will change their own y positions accordingly as well.

In our project we didn’t use any external JS library, we solely used base javascript. Because of this, I felt as though we met the requirements for our given tech stack. We used ES6 classes, modules, and transplied our code into ES5. I feel as though our code is well commented through and through and I deleted all of the console.log() calls. We did borrow code for delta time from: “<https://isaacsukin.com/news/2015/01/detailed-explanation-javascript-game-loops-and-timing>”.

For the sound function in the main.js, I used a tutorial found on w3 schools to help me. The link for this is:

“<https://www.w3schools.com/graphics/tryit.asp?filename=trygame_sound>”.

For the most part, a lot of our project went according to plan. However there were some things that we had to accept and cut our losses on. We were not able to fully animate the ship or give the bullets particle effects as we had hoped to to do. During this project, I would say that I did majority of the work for the first half of the project, this includes making the sprites and background as well. Zach was able to majority of the work for the latter half of the project. With that being said we both would help one another regardless of who was primarily working at the time. I would say that the workload in this project was split pretty evenly between the two of us.