Taylor

ART385

Code Self Portrait to P5.js

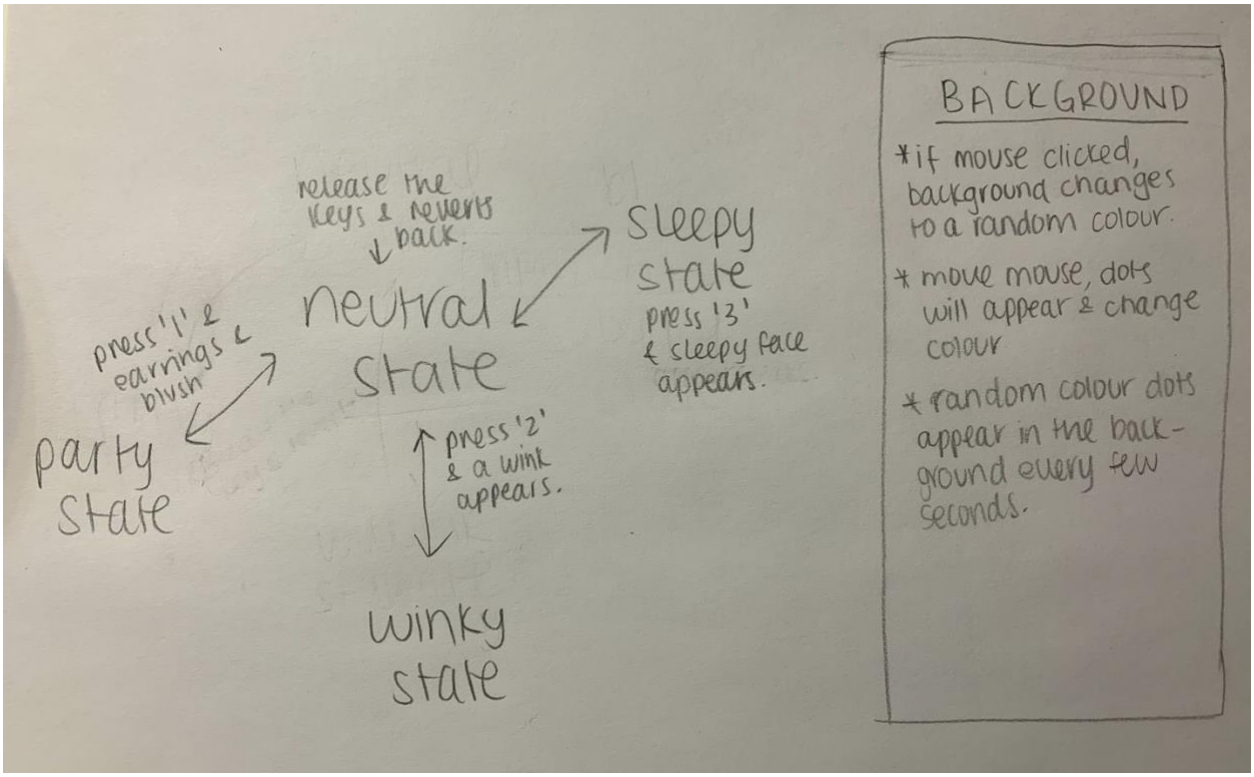
February 13th, 2020

Code Self Portrait to P5.js with Feelings

The Assignment:

For this assignment, we were meant to import our coded self-portrait from the Processing software into P5.js and implement interactive states that would show different emotions by pressing keys or moving our mouse around.

Interaction Diagram



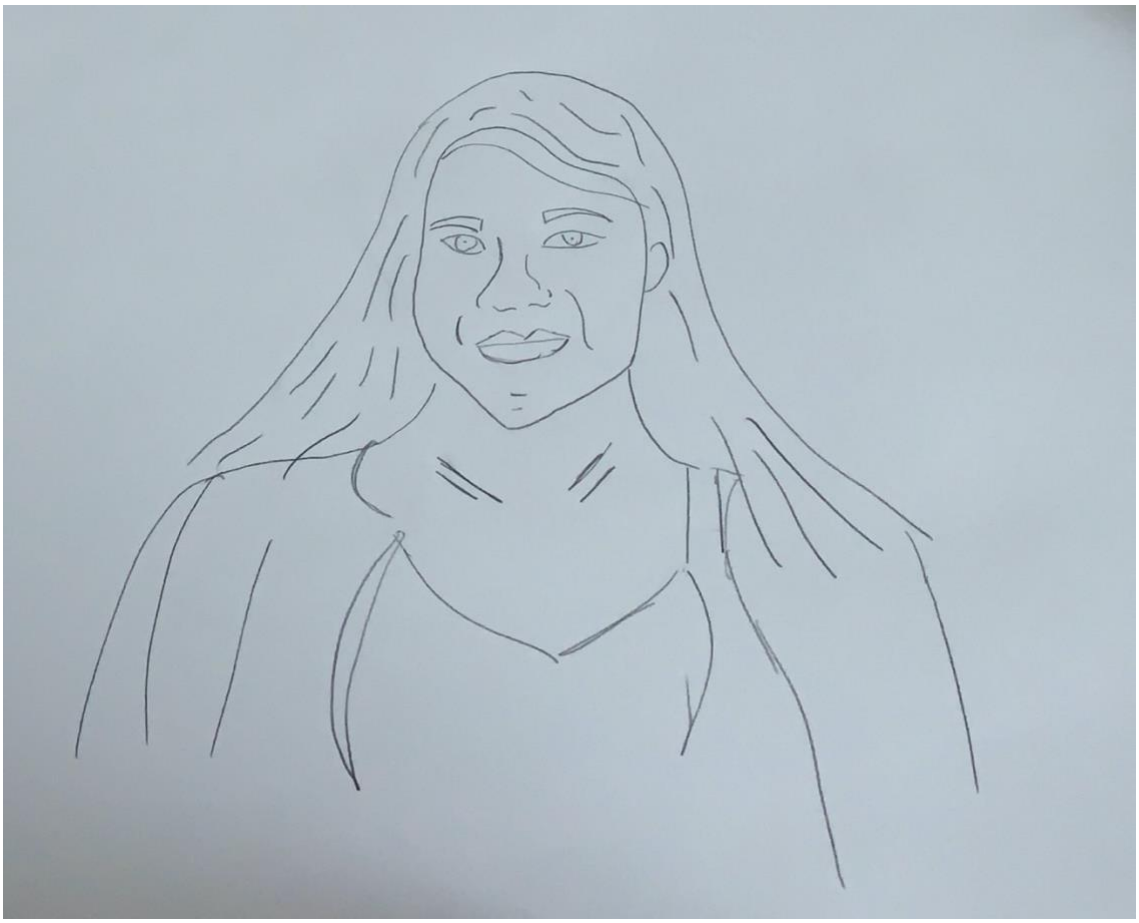
Taylor

ART385

Code Self Portrait to P5.js

February 13th, 2020

Hand-drawn Sketches:



Reflections:

For this project, it was interesting as our coded self-portrait written on Processing got to be imported into P5.js, a new programming language that I got to learn and use. I found it quick to learn as most of the syntax was the same except for the following, in Processing and other programming languages to define a variable, you use ‘int’ to initialize it. But in P5.js, you use

'var.' Another example, we used 'function' instead of 'void'. This assignment really helped me understand P5.js, and it was fun to implement different states using keys or moving the mouse or clicking to ensure an interactive design.

Moreover, I made three different emotional states: a neutral state, a 'party' state and lastly a sleepy state. The neutral state was my static state and would continue to revert back to that stat after pressing keys. For example, when pressing the '1' key, earrings and a blush would pop up on the person and then would revert back to the neutral state. Similarly, when pressing '2,' I added a wink where one of the eyes stayed normal and the other one would change. When pressing '3,' the eyes changed to look down and my mouth would change as well to resemble a 'sleepy' state and then revert back to the neutral.

Lastly, I found this task to be a lot easier in terms of creating and coding the portrait than the first as I knew what I was going to design and had a clear outline of what I was going to do. However, coding with P5.js became the struggle as I had a lot of debugging to do and had to restructure my code, in order for it to run smoothly. This is because, at first, I was only able to implement one key to work, and when I pressed the other keys, my design would freeze and stop working. I also had issues with changing the states and the background which was frustrating until I realized that I had to create different functions for each state to work and for the background to continuously run. Nevertheless, I really enjoyed this task and it was cool to create an interactive piece!