Jeffrey

ART385

Simple State Machine

Tues, March 24

Re-state the Assignment
Make a state machine that controls your LEDs. Make an LED turn on after pressing it 4 times and then turn it off after another 4 presses. Then try to make your own interaction.

Hand-drawn sketches
https://drive.google.com/open?id=1YvoW2c8vkZkwdLp2KNBuuQR58WvjybiT

Reflections
The state machine prompt used a lot of the RGB LED code from the previous weekly assignment. This time, instead of using dutyCycle to change the intensity of the light, I just kept the pulse at 255 and used the different colours (Red, Green and Blue) as different states for the state machine. As such, the state machine will have 4 states: red, green, blue and off. The code for this project reminded me of a slideshow app I made in 107 where clicking a button will cycle through images at different indices before resetting to the first index. As such, the conceptual part of this assignment was easy to grasp. However, coding the part where it recognized individual button presses was extremely difficult. I realized that the delay function would help the program differentiate whether someone is holding the button vs clicking but this method was too static in my opinion. As such, I tried using a boolean at first called ifPressed and changed it to true whenever the button was pressed as well as a conditional that prevented that only incremented the number of clicks once as if "ifPressed==true". This did not produce the results I wanted and this became a pretty big wall for me during the project. Thankfully, this was covered during tuesday's lecture so I was able to use the method of comparing the current state vs previous state as a way of differentiating a click from a hold. Overall, this assignment was neat. It gave me more insight on the fidelity of codes. Hopefully, I can look more into delay though.