Bryan Xu
CC Fall 23
Final Self Assessment

• Critically analyze/evaluate how much time was spent learning syntax & structure, programming concepts vs. actually programming, and how does this reflect on the final quality of your end result.

I feel like i've spent a lot more time doing actual actual programming than learning the syntax, and I don't think I got to spend that much time on programming concepts either. I think at the end, I would've benefit more from learning more programming concepts throughout the semester just so that it would help me to get a grasp on a more of what I'm doing in my final project. Right now, it does feel like I was learning a lot of stuff that hadn't done before just for the code in my final project; it was definitely beyond the scope of content I was familiar with prior.

• Comment on your successes and frustrations with Processing and P5.js.

As for successes and frustrations with p5, I feel like the it was definitely a lot of frustrations to get things working on p5js and also porting them over to open processing. I think it has a lot to do with how using external libraries work, and then sometimes the lack of indices, the lack of references made it difficult. It was very hard for me to look up the functions that I wanted to execute, and I would have moments when I feel pretty clueless about what I'm doing. I do think I amazed myself to a degree when I found out how I'm able to produce things that look so visual and/or interactive just with code in a generative design more and more so throughout each sketch, and this is definitely a new opening-up for me.

• Compare and contrast OOP versus Procedural Programming. How are they similiar? How are they different? I am looking for you to explain this in your own words, what these concepts are. I am NOT looking for you to tell me how you used these in your project.

My interpretation of OOP vs Procedural Programming is that OOP is much more real-world like. This is in the sense that the real world is very much also based on objects and their unlimited forms of interactions. In the same comparison, Procedural Programming feels much more like some vehicle that gets me from A to B, executing some specific function. There's this very slight tipping point where if i start to think of functions more as tools that can apply to a variety of purposes, I feel like I'm leaning towards an OOP type of thinking, where it feels much less sequential and more so like a big web of things that may take to all types of places.

• Specifically considering your final project: What programming concepts solidified in your final project? What did you learn with reference to programming? Did you have a break through?
• Specifically considering your final project: Were you able to resolve your own bugs? What tricks did you learn in the process to help? Did you do any debugging?

Regarding my final project, I think one of the most reinforced concepts that I've understood through the project has to be setting up flags and printing out statements in the debugging process because my project involved a lot of that. I would also say that I gained a lot of experience on setting up programs to be more expandable because that's what I wanted to do: I wanted to have a lot of different levels that popped up. That vision didn't turn out to be true; however, I still followed those general principles when I was designing the basics of my classes where I would have triggers, update methods, and also draw methods for each class (at least that's what I intended to do). I think this is my first time really trying to set up a program to be truly expandable (polymorphic in a sense), so it was definitely a valuable experience despite the end result.

• What are you most proud of, with reference to your final project?

I think that the aspect about my project that makes me the proudest is probably managing to salvage my program from endless types of bugs every time I broke my own code trying to do something new/more. Getting the noise detection and drawing functions ironed out and working was actually a lot of work, it was way more than I thought it would ever be.

• What was your intended milestone? Did you make it? Did another one pop up? Tell me about this. Tell me how you resolved it.

My original milestone was to have ml5 working as a way to use hand movement to simulate the motions of swinging an axe that my character was supposed to perform. But then I realized at the rate ml5 was refreshing, it would be way too slow; so I gave up on that idea. I still did want to try make ml5 work in the capacity of recognizing how spread out my hand is and manipulating that data, but I unfortunately ran out of time trying to implement it again; it just never worked on me.

• How do you think you'll move forward with programming? will you keep doing it ? How does this relate to other classes you are either taking or wish to take?

I guess I found coding to be very useful in these output, type of situations where you have to execute an idea Through this class. In the future, I think I'm definitely better equipped for scenarios that need me to code, but I also feel the sheer sense of limitation a lot about how programming doesn't really flow as naturally to me as some other things. Still, I appreciate the solid training I gained from this class and how it reminded me that I definitely CAN learn how to code.