Creative Coding Final Self Assessment

Julia Chen

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 spent a decent amount of time learning programming concepts. Seeing examples helped me a lot when I would start actually programming. I would have some ideas in my head about what I wanted the final output to look like, but it was difficult to remember what the actual syntax was, and that required some more research and learning.

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

- I haven't really used Processing since around a few weeks before the midterm, so I can't comment on recent experiences with Processing. However, with P5 I felt that the sky really was the limit for what I could create. All the libraries we have access to with P5 made it easy to implement a lot of cool things that I hadn't even considered before. Processing is definitely a more straightforward and intuitive application to use, since it doesn't require the whole process with setting up the index file and more. I did have some difficulty working with Sublime for my final project. I spent hours trying to figure out how to set up the Node server, and I ended up deciding to just focus on making it work directly in OpenProcessing.

Compare and contrast OOP versus Procedural Programming. How are they similar? 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.

- Procedural programming is much less flexible than object oriented programming. If you're using procedural programming, you're telling the computer exactly what you want it to do, line by line. Both with both object oriented and procedural programming you have parameters that are somewhat customizable, to different extents. Object oriented programming, you're putting information into objects, but with procedural programming you might be passing information through functions that you've created.

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?

- The concept that I really solidified in my project was using APIs. Although I had some issues with the website I was using and it caused a great deal of stress and anxiety, I did so much troubleshooting with APIs that I feel like I have a pretty solid grasp of how to use them.

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?

- For the most part, I was able to resolve my own bugs. Many times they were just silly mistakes, like when I just forgot to call the function Other times I asked my classmates for advice

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 milestone was honestly I was a little ambitious. I wanted to implement every single state into my finished product, but it would have definitely taken much longer. I decided to change my plan and focus on a few states, then polish those screens. I was successful in making the individual screens

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

- I'm proud of how I combined a lot of concepts that confused me for a while in my project. Libraries and APIs were really daunting at first when we first started talking about them. Using libraries opened up a lot of possibilities for me for this project, and I feel like I could definitely take things one step further if new ideas come to me in the future.

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?

- Programming is a pretty important skill to have, so I definitely will continue to learn it. I might try learning different languages in the future as well, since different languages are applicable for various lines of work. I think learning C/C++ might be really useful for me in the future.