Sophia Ferrera

Katherine Bennett

Creative Coding Final Deliverables

17 December 2019

Throughout the semester, a majority of my time coding was spent on understanding syntax and actually programming after learning concepts. Structure was not necessarily a big concern of mine based on the fact that I had prior coding experience, but I did spend a lot of time trying to understand syntax and why certain concepts are formulated the way they are. Vectors, for example, were a lot easier for me to grasp because I could connect it to my prior knowledge of physics, but learning classes in P5.js was difficult when we started to try to break down "this." and what pointed where. The time I spent trying to grasp classes in p5.js reflected on the final quality of my project because when it came time to begin coding it, I could easily tackle what I already knew how to do so that I could begin to manipulate the P5.Play library to make my project more visually interesting. I think it took me a long time to grasp many concepts in terms of applying them to my own code because I was looking at them in a very objective way. I would try to grasp the concept as it was taught and did not understand how it could possibly apply to my own code. I understood that I couldn bend the code to my will, but I often found myself neglecting to stretch the possibilities of these concepts enough to solidify them. I can't say I'm 100% on every concept we learned, but I definitely feel more capable of refining these skills than I did at the beginning of the semester.

My successes with Processing and P5.js in the beginning were minimal, making my midterm self-reflection a pity party for myself. If you had told me at the beginning of the semester that I would end up making the final project I produced, I would not have believed you. I had a lot of confidence in myself in mostly all of my other classes except when it came to coding, so I found myself discouraged by my past failures instead of driven to learn from them. In terms of the languages themselves, I think coding itself is so finicky and specific, but completely rewarding. My successes came later on and from a place of patience. I think if I had learned what pseudo-code was earlier on, it definitely would have been helpful to break things down. One of the most valuable things I learned how to do was to break down my code into the smallest functions and tasks possible so that debugging and even conceptualizing became easier. I'm not sure why, but I prefer P5.js significantly more than Processing. It might be because we learned about libraries when we were learning P5.js, but I feel a lot less restricted by the possibilities in P5.js. I am very much still frustrated by it because it is the perfect example of an interface that will do *exactly* what you tell it to, which might mean that the way I chose to execute something may end up backfiring when I try to tackle more complex concepts.

Object oriented programming and its use of classes make it a useful tool when coding multiple variations of objects from different classes. OOP, like described in class, is like using a cookie cutter. It will make the same shape (i.e. it will be the same class/structure of an object), but the flavors (i.e. the inputs of each parameter) will be different. This adds a significant amount of ease when programming many different objects from different classes since the code can be adjusted without much alteration of other code from other classes. Procedural programming can also contain functions that are equivalent to the methods within a class, but it becomes much more difficult to manipulate the code later on due to the fact that it is programmed in accordance to how it should function (i.e. the procedure it should go through) without much room for malleability.

In my final project, I definitely solidified the use of classes, for-loops, layering, interaction, and switch statements. In reference to programming, I learned how to utilize the P5.Play library, how to use a switch statement, how to create animations, and how to use the P5.Sound library. I had a couple of breakthroughs along the way, beginning once I simply figured out how to display my animations. When I began to see what I had been visualizing, it made me think more about how to add interest both visually and audibly in my project. I also had a breakthrough after a bit of experimenting with my switch statements. This is not a hard concept but I was really proud of myself when I figured out how to apply it to my own work. There were rules regarding syntax I was not following, but when I learned what could be manipulated and what had to exist in order for it to run, it refined the way my code ran as a whole. It was also a very fun project to create in general, and I was genuinely excited to sit down and work on it.

In terms of bugs, I was able to resolve a majority of my own bugs with some occasional conceptual help from friends who study computer science. Most of the time, my bugs were things that were a result of me not keeping track of my variables correctly or not breaking my code up into small enough pieces to understand what was happening. To help, I make sure that basically all of my functions were doing one thing at a time. I would run the background, animations, objects from classes, and sounds all separately to make sure they were working on their own. I then proceeded to combine these concepts, and as bugs arose, I could be sure that my individual functions were not the problem. My biggest issues were really only with my switch statements for my sound files, and these problems were resolved once I figured out where/how to keep track of my counting variables and when/where to reset them.

Moving forward, I'm not sure what my future is with coding. For my major requirements next semester, I am taking an introductory Python course. I am not really looking forward to problem solving after having a semester of being able to creatively code, but I think I can go into it with less trauma and doubt than I would have if I hadn't taken this course first. I can't see myself using code as my main platform for design, but I am really glad I have enough experience to be able to say that I would definitely be interested in continuing to grow my skills. I can see my skills in this course being applied to WebDev or other UX/UI classes required for IDM. I can see myself potentially taking a course on data visualization in the future, but I'm going to try and get through my Python course first.

My piece "Road Trippin'" is a digital experience that simulates a fantasy road trip consisting of different scenes and accompanying ambient noise. I wanted to create a digital art-based piece that would communicate a lighthearted tone where the user could "tune" the radio and experience the changing scenes. It is both an interactive and time-based experience that allows the user to put themselves in the place of the driver or passenger. I ultimately did not choose to change anything after presenting because I felt more confident in the functionality of my piece when both the sound and the visuals could be controlled. In my first critique, I found that people preferred having control over each situation, making the piece less of a narrative.