Final Deliverables_Aubree Stephens

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.

- For the final project, I spent about 50% of my time learning syntax associated with the Scene Manager library and researching existing examples of code that work with text and images.
- Understanding the syntax of the library ultimately made it easier to organize and replicate my code for each city. However, there was a pretty steep learning curve initially. Consequently, my visuals appear much simpler than I would like them to; it feels like the majority of the work is "under the hood."
- Reflecting upon my process, I spent far too much time looking for pre-existing, sophisticated examples of code and not enough time implementing the foundational aspects of code, in a unique way.

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

- I was initially very frustrated with p5.js, GitHub and libraries. However, once I started using them, I realized the importance of each and developed a respect for JavaScript.
- I continue to be frustrating with the reliance on Internet to successfully work in p5.js. I, in no way, enjoy having to open my desktop project folder, locate my index.html, and open in Firefox in order to run my program. However, I recognize that this is a small price to pay for my code being "universally accessible" without reliance on local software.
- Processing encouraged me to work on my project "on the go"- some of my best ideas/ lines of code came from my hour-long commute.
- P5.js, on the other hand, encouraged me to be more organized with my libraries and folders. It also encouraged me to intentionally carve out large blocks of time to just sit and code.

Compare and contrast OOP versus Procedural Programming.

- ***Object Oriented Programming***
    - Object-oriented programming relies heavily (at least initially) on a solid understanding of procedural programming. Once you've worked through all of the logistics of eliciting a certain

response/result, you can take a step back and ask yourself “is there an easier, and more efficient way of doing this?” The answer is usually yes, and often requires the creation of a class and subsequent creation of an object.
    - Once a programmer is more advanced, they can certainly *begin* with object-oriented programming- just as one would choose multiplication over long strings of addition. Until you are comfortable with the skill, you should outline what you’d like to accomplish, procedurally.

- ***Procedural Programming***

    - My overall understanding of procedural programming is the idea that *most* sketches require an initial set up and a draw loop- what you want to initialize and what you want to actually *do.* From there, you can set up functions, which really just organize groups of tasks. Subsequently, you can declare variables, especially for things that are called upon multiple times, or things that change over time. More than likely, your code will become more and more complex, and will encourage you to consolidate/reorganize into object-oriented programming.

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?

- I utilized object-oriented programming with regards to my “city.js” sketch. This allowed me to display the city name on the map when moused-over. I was able to create constructors, pass parameters, and utilize dot syntax. Overall, I utilized JavaScript language with greater confidence. I used functions where applicable and I declared variables wherever possible.
- I learned how to incorporate other libraries into my project. I learned how to import custom font. I learned how to organize and preload multiple images. I learned how to edit image size in “System Viewer.” I learned, in greater depth, how to use mouse-over and keyPressed functions.
- I think my breakthrough was passing parameters- I finally get it! It’s extremely useful; I just needed to practice.

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?

- Half, if not three-quarters, of my project has been spent debugging. There was a very challenging transition in utilizing the scene manager library. It was worth the work- once I figured it out- but for a long time, I could not figure out the syntax it wanted me to use and it wasn’t well documented, in my opinion.

- With regards to debugging, I used the “K.I.S.S.” method- keep it simple stupid. If I couldn’t figure out why a bit of code wasn’t working, I’d comment out all of the nonessential parts and focus on the simplest execution of what I was trying to accomplish. I also talked through my code. This helped me when I was placing certain items in the setup function when they really should have been in the draw function (and vice versa).

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 will certainly implement my understanding of programming into my high school engineering curriculum. As you mentioned, time and time again, “you can do anything with code.” I want to impress this upon my own students.
- In my “Developing Assistive Technology” Course, I designed a game that encourages sound localization and discernment in children with auditory processing disorders. Currently, the game requires that the occupational therapist play sounds to four different speakers by manipulating two applications, at the same time- Airfoil and iTunes. This is cumbersome and I would *love* to create a user interface that is much more intuitive for the occupational therapist. I hope to bring this to fruition, next semester.