Nandita Kohli

14 December, 2018

Professor Bennett

Creative Coding

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.

Overall I spent more time on research and programming concepts in the second half of the semester. This helped me get more of an understanding for the syntax and concepts of programming. I think the structure came more from coding trial and error, however. The research really improved my end result.

My midterm project was more coding than research, and although it turned out well, I think my final project was more complex and fleshed out thanks to the more extensive research and debugging I had to learn. While the midterm project introduced me to thinking in terms of coding projects, the final really solidified my learning and understanding of concepts and taught me more on the complexities of coding larger scale projects. With knowledge gained through the final project coding itself became easier, more interesting and more enjoyable.

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

In the first half of the semester I used Processing more. In the second half of the semester I used P5.js more. I struggled to understand how to open P5.js in just a web browser and edit code straight from sublime and save it. This was always somewhat confusing to me. However, OpenProcessing made it much easier for me to work with P5.js. This was great because I got to see the differences between the two programs. I got to see how javascript differs from java. Javascript seemed a lot more flexible. You could do a lot with it and it was slightly less picky about semicolons and such. I decided to use it more second semester because it seems to be far more used overall and I wanted to get in more practice.

I think overall it paid off and now I feel generally comfortable to explore both. I think I also got pretty far in both. I did my midterm in Processing and my final in P5.js. I think overall, P5.js is better for web stuff and animations and processing is better for hardcore art and design. Both are

quite nice, though, and have really helpful tutorials. I'm glad I started with java and then transitioned to P5.js though.

Compare and contrast OOP versus Procedural Programming.

Procedural programming (POP), admittedly makes more sense to me. This is probably because I am slightly more familiar with it. It is the step-by-step approach to break down a task into variables and functions. Each step is carried out in order. Object oriented programming (OOP), on the other hand, involves classes of objects and methods. OOP focuses on data rather than the algorithm, while procedural programming focuses on procedural abstractions. In OOP, the program is divided into objects which are instances of classes. In POP, the main program is divided into functions.

I understand the importance of OOP and want to continue to improve at it for the future. I think it makes really large scale screen animations much easier to work on.

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 had quite a few breakthroughs in the final project.

For one, I realized that there really is no reason to fear what I don't know or fear 'messing things up' by trying new things. Everything is on google these days and as long as you make multiple drafts, you should be fine and not worry about tinkering with your work.

I solidified my understanding of APIs and external libraries in the final project. I now really understand better what an API is and what an external library is. I understand how they function and how to work with them and where to start. The start was very confusing in the beginning. I had to first research what these elements were before really diving in and understanding how to start.

In addition, I learned about structure and how to best load and manipulate data. In the beginning one of the issues I had was with not understanding what to put in the draw loop versus the setup loop. I also didn't understand exactly where functions go for the best structural practices. Now I understand this way better. I also solidified my understanding of loops and variables. I understand how to work with variables more and also use variables inside of loops to trigger changes based on data specificities. Dot syntax was also an important part of this as well.

I also now understand how to parse through data in the console and how to parse through code without feeling as intimidated. It makes so much more sense to me now. Anything I do not

understand I am also now more comfortable just googling it to find what I am looking for. I feel more prepared now to tackle my creative ambitions and use the tools available to me.

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?

I did some debugging. It was tough though. Lots of my issues revolved around not understanding how the console worked, structural issues, and confusion working with external libraries. I could not figure out some of the issues, however, and this is when I decided to visit the coding TA who was very helpful. As I watched him, however, I learned a lot about debugging and how it works. I learned that googling really is your best friend. I learned which sites to go to for the best advice and which parts of the sites to look at. I also tried watching videos by Dan Shiffman. GitHub is another good resource. The console is very helpful as well. I also learned that it isn't enough to just visit the console. In the beginning I would visit the console and get intimidated by the sheer words on the page and felt like I understood nothing. As time went on I learned how to read the console much easilier. I would go to the console and then follow the links in the console to figure out what was going on. And if worse comes to worse, googling what is in the console is a very good start.

In the beginning I was frightened by the error messages and the fact that I did not know what the problem was. Then I realized that that feeling is totally normal and in order to figure out what is wrong everyone would generally have to do the same thing: check the console and then google it. This was somewhat of an ironic epiphany.

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 really enjoyed this class. It was definitely a struggle, however I feel as though I learned a lot of skills that will all definitely come in handy in the future. Beyond simple programming structures, this class really taught me how to accomplish large projects over spans of time, how to never give up or get discouraged easily, and how to find solutions for difficult, confusing problems with minimal straightforward assistance. It also taught me how to just have fun with code and be creative and not get intimidated by the enormity of the process or the unknown.

I definitely hope to keep improving my coding skills over the next four years and beyond. I already have a lot of ideas on how to improve my final sketch that I want to try implementing. In addition, I am taking a computer science beginners course next semester. I hope to also get a chance to take 3D modeling in the future.

Lastly, something I really enjoyed about this class was the amazing balance between creativity and structure. I was able to learn a lot of technical skills while also making projects that really

used and showed my own creative thoughts and interests. I was really amazed by all of the final projects in the class. I hope to find more classes like this -- with this balance of creativity and structure and skill.