

When I first started computation art, I only knew basic python, HTML, and I had tinkered once with Arduino, so truly at best I was an advanced beginner. As someone who was self-taught, I only had my partner to run to when something wouldn't work or internet forums. While I don't always know when the answer to everything, I now have the ability to efficiently trouble shoot (debugging) most problems I come across when coding

At first, when I read the course outline and saw we were going to use JavaScript, I started panicking. I thought of starting the course for JavaScript on Codedex to get a head start because I thought it might be too hard. So I was strangely surprise of how certain rules I saw in python also applied to this language. For me, else if/if was something I genuinely loved doing in with python, once I understood it I used it everywhere all the time for anything. It took me by surprise though when others would look at my code and point out the excessive use of else if/if. While I was getting taught new functions and concept such as variables or arrays, out of fear of stepping out of my comfort zone, I was sticking with the old stuff. Learning from others and collaborating with classmates on certain projects made me realize that we are all figuring new things out. As I stepped out of my comfort zone, I had resources such as debugging videos to help me once I hit a wall. I had classmates and friends I felt confident asking for help, and that made me more confident to apply new knowledge.

I noticed at the start of the semester, that when people asked me about my portfolio submissions, I wasn't able to explain to them. Not only that, but I wasn't able to explain how the project was made, why it was made, etc. I could only show them and then explain because, as an artist, I am often used to describe my work in an emotional and descriptive way. Considering most of my portfolio work consisted of HTML, blender, Maya, Arduino, I found it hard to stick to my good old ways. As classes went by, and we had moments such as thinking like a programmer/computer, I started being more comfortable explaining my work to others and myself. I knew how to explain how it was programmed, what the goal was AND still make it descriptive and emotional if need be! I find it easier to explain and people actually understand what I am doing with my code.

Prior to this semester, I was following classes on codedex on some programming language. Because they were divided in section, I often worked 2/3 weeks learning and working one specific language. Since I started this semester, I gained knowledge in different programming languages I saw how they are connected with each other. I found it funny to see I could use js5 for a website. I didn't understand fully that they were all connected at first due to my prior self-taught knowledge and how it was taught. That made me realize I shouldn't try to master JS5, but rather understand how to use and how it works. As an artist, we often try to master what we are considered less good at. If you suck at drawing feet, you will spend time mastering feet if you want to get better. Granted, the rest of the body components aren't being worked on but drawing only feet

is considered art. As a creative coder, if I suck at HTML, the need to master it isn't as important. I rather work with the language by having a couple of project where it also integrated, but I won't obsessively work only with HTML. I work on a project, and suddenly I need to use HTML for a certain thing and I start using my solving problem skill to see how I could make it happen and my past knowledge, worse case I'll google it, but to thinker in all languages no matter the level of mastery is something I try to incorporate more and more in my own artistic practices.

One thing that did sting me was when I lost points on my comments for the self-portrait. I remember thinking :‘Well it's my code I don't need others snooping and looking at it’ and because artist work is only collaborative when agreed upon, I didn't understand the need to comments in detail if I knew my code. It still is something I have a hard time doing as I program, but I do find it easier to take time after to thoroughly read my code from top to bottom and add comments where it is needed. I see its usefulness for my own person. I still make them personal to me, they are like journal entries and reminders for the future, but I know that they are still understandable to others. While I find it weird to have others read and access my code, I feel myself less ashamed in the way I express myself even in comments.

I also enjoy silly and simple thing. While program, it's hard to feel like I'm using my creative capabilities in most class. It often depresses me a bit how little creativity was promoted in cart 211. That pushed me to make it my mantra for each of my assignment or project to be an art piece. My website was a magazine – a written art form of art. Frog game become a story. Instead of seeing my work as just a code or a website, I transform it into a form of art I can still pour my logic and creativity into. Learning to combine both sides helped me identify a bit more with the label creative coder. Despite that, I know I will mostly label myself as an artist.

For my future,

I think I need to immerse myself more into the creative coding scene. As a visual artist, I would go to museums, expositions, movies, and proceed to get inspired by these. As a creative coder, it is a bit harder to find inspiration beyond what I see my peers doing. I want to see more work from creative coder fully integrated as an art piece outside of academics settings. And so as a creative coder, I have to look more into things that fascinate me. When I like someone project, I rarely look at their code. I need to take time to actually look at it. Everyone is borrowed from everywhere so I shouldn't feel ashamed to look at the inner work of a project because I respect it.

Being a creative coder is new to me, and I am still navigating what it means for me. It is a blossoming relationship between me and my new-found art direction.

