Reflective Essay

CART 253 AA

Noureddine Mazzene
Sabine Rosenberg
December 8, 2025

When I look back at where I started at the beginning of this course, my relationship with coding felt distant and almost fragile. I had touched code before, but only in very controlled environments where everything was guided step by step. My previous experience came from an introductory web design class where we focused on HTML and CSS, and even then, our hand was held the entire time. I learned basic structures, tags, and styling, but I never had to face anything that felt like “real” programming. Anything involving logic, equations, or problem-solving seemed intimidating, and I saw JavaScript as that one big wall that “real coders” dealt with. Because I came from the DART program and not CART, I already believed I was entering the course from a weaker position. DART felt design-oriented and laid back, while CART always looked more technical and intense. So when I walked into this class, I felt curious but also unsure of how I would handle something that seemed so tied to math, which I never felt particularly strong at.

Even though I didn’t think coding was “not for me,” I didn’t see myself as someone who could actually use it comfortably. It felt like a world I admired from the outside or one I could talk about with friends who code, but not a world I belonged to. I didn’t know what JavaScript had to do with creative work, and I definitely didn’t know what p5.js was. The moment I discovered p5, though, things shifted. It took something that looked complicated and made it feel more approachable. Understanding that p5 is just a library layered on top of JavaScript helped me reframe the process: maybe the “big scary language” could be broken down, and maybe this course wasn’t going to drown me in equations after all. Still, diving into the actual code early on, especially when motion and interaction were introduced, brought the intimidation back. That was the moment when I realized that this wasn’t just HTML and CSS anymore. What helped me most in the beginning was learning the basic logic behind programming. Variables, loops, and especially if-statements began to feel like the backbone of what coding really is.

For some reason, if-statements felt the most familiar, even before taking the course. “If this happens, then do that” matched what I always imagined coding to be, and finally understanding how it works made the whole structure feel less mysterious. Arrays also opened a huge door for me, because suddenly I could store multiple things at once and rotate through them. It gave me breathing room in my code, like I could finally organize thoughts instead of stacking them in random places. Throughout the semester, these technical concepts stopped being abstract ideas and became tools I could actually hold in my hands.

The project that changed everything for me was the Mod Jam. That experience woke up a memory from when I was younger and used to mod Super Smash Bros. Brawl. I would open tools that let me change text colors or import new textures I made in Photoshop to create custom costumes. It felt like hacking back then, or like opening a hidden layer of a game that only a few people ever see. Working on the Mod Jam brought that feeling back. Editing the frog game felt familiar in a way that creating something from zero didn’t. I understood where to look in the code when I wanted to make a change, and it felt like I had a map to follow. Modding let me step into coding from a place of comfort, and I honestly think I would have been five times more intimidated if the course had started with building everything from scratch. The Variation Jam continued that same momentum. It felt natural to build on top of something I had already shaped my own way. My biggest changes throughout both projects leaned towards visual design, which reflects who I am as a designer. I would first think of the assets I wanted, create them, and then go into the code to make them functional. That process helped me better understand the relationship between designers and developers. Designers produce the ideas and visuals, and programmers figure out how to transform those ideas into interactive experiences. Working with p5 made that connection clearer than ever. I started to understand why design and computation arts exist as two separate programs, but

also why they still overlap. Coding influenced the way I structured my work, forcing me to think in steps, dependencies, and systems rather than just visuals. It made me aware that creativity doesn't disappear in code, it transforms into logic, relationships, and movement.

Even though I've gained a lot, I still have challenges that follow me. Debugging continues to be the part of coding I dislike the most. One small typo or missing symbol can break everything, and every time something goes wrong, my mind jumps to the worst-case scenario, like I ruined the whole sketch. Then I find out it was one extra space or a missing bracket. Syntax is another thing that still gets in the way. I often feel like I understand the concept perfectly but lose confidence because of the technical precision coding demands. It's frustrating, but it's also a normal part of learning how to work with something so exact. The difference now is that it doesn't feel like a mystery anymore. Coding used to feel like this world with a hundred locked doors, and I didn't even know where the keys were. Now it feels like one huge wall I just have to climb, with tools I actually know how to use. I don't think I fully see myself as a "creative coder" yet, but I'm much closer to that identity than before. When I open a blank p5 sketch, part of me still hesitates because I rely on templates, but the other part knows exactly how to navigate. I know where to start, what structure needs to be there, and how to build from it. Most importantly, I feel capable of figuring things out on my own. I know how to use documentation, how to import my own assets, and how to translate a visual idea into coded behavior.

Looking ahead, I'm excited about the independence that coding opens up for me. I can see myself applying this knowledge in so many directions: web design for my own agency, interactive projects, app design, or even taking steps toward making a game one day. Making a game has been something I've dreamed about since I was younger, and every time I learn more about programming, that dream

feels slightly closer. It's not that I suddenly mastered everything, but I now have a foundation I can grow from. The more tools you have as a designer, the more possibilities you unlock. Photoshop is second nature to me, but adding code into my skill set expands what I can bring to life and how I can bring it to life. The next step for me is learning vanilla JavaScript outside of p5. I appreciate p5 for how accessible it makes the learning curve, but I also feel like understanding JavaScript directly will help me break into more "industry-standard" environments. Most people I know outside of school have never heard of p5, but they all know JavaScript. That gap makes me want to learn it more seriously so that I can stand on my own without depending fully on a library.

Overall, this course changed the way I see code both technically and creatively. It made me realize that programming isn't a world reserved for math geniuses. It's something that opens itself to you the more you experiment, break things, fix them, and shape them into something you enjoy. I entered this course intimidated and unsure, but I'm leaving with a clearer sense of how coding fits into my creative practice and where it can take me in the future.