Dec. 4th, 2025 Nerly Cadet

My programming knowledge at the start of the course was not just small, but non-existent. In fact, I had never learned or studied any coding language before, much less JavaScript. Since I was enrolled in a social science program in college, I was actually unfamiliar with both art and computation as a course. On top of that, many of my classmates from CART253 or other CART classes had backgrounds in coding, either from college, from their actual jobs or from individual learning. I was a little anxious about this at first, since I thought I might fall behind, fearing feelings of imposter syndrome. Additionally, because computer science is a subject that is frequently thought of as being extremely difficult to understand and do, I was initially a little afraid. Later on, I was able to overcome that nervousness really fast, thanks to the way the course was structured. The pre-class films were very helpful and educational. Additionally, because it was taped rather than in person, it allowed me take as long as I needed to listen. I was able to slow down or speed up the video. Skip parts I understood or rewind back to things I didn't fully comprehend. If it weren't for the videos, I doubt I would have grasped some of the things that I now understand. The majority of my anxiousness subsided in class after I noticed that I understood most that was being taught and knew exactly what to do during the group challenges. This made me feel intelligent and confident about coding. Additionally, my teammates and I were able to rely on one another and learn from each other's failures and additional information because we were able to work together amid problems. I would continually urge her to commit. We could always count on Ashmitha to provide clarification, and Ya Xuan had extensive coding experience from her prior college courses. Because I missed certain weeks due to either procrastination or overwork, what I currently know is a little less than what I should know. Because of this, I tend to be more

confident with the aesthetic aspects of JavaScript coding, rather than the interactive aspects. In reality, I frequently conduct extensive study when it comes to the interaction of my codes. It's not that I don't comprehend the ideas; I just don't always know how to apply the language properly. Even though we were taught to think like computers, group thinking always seemed to make things easier, because it allowed me to question things I normally would not question if I was doing a challenge by myself. Take if-statements, for instance. I've been getting better at them lately, and I'm almost ready to use them without browsing the internet, the p5 library, or my files to use previous projects or challenges as guidance and references. However, I'm still having trouble deciding what to include and how to write inside these statements to make my idea come to life. However, when that concept was thought through videos, and explained in class prior to studio time, everything was crystal clear to me. The maths is no different. I've never been good at math, and even though JavaScript math is quite easy, it's the thing I dread the most, such as vectors. Using width / 2 to position an object. Having greater than, less than symbols. And to be honest, the whole entire concept of for-loops. The issue is that reasoning behind it makes sense to me when I watch the videos. When doing challenges in teams, I still understand the logic behind it. However, I find it challenging to know when and how to apply them when it comes to my own personal Jam projects. Despite this, I think that my present comprehension of codes when reading has been the most significant change from my past coding skills. I am now fully aware of the distinction between mousePressed and keyPressed. I did not before. I am now aware of what goes inside the function setup and what goes inside the function draw. I see the reasoning behind that, but I didn’t before.

Despite all these shortcomings of mine regarding coding, debugging is one of my strong points. I always know what to look for first when searching for a bug, even if the console does not display the problem. I do not panic when I use console.log but am unable to locate the problem. I know to prioritize my focus towards a possible undefined function, a function with no closing bracket, or a misspelling. It can be anything, and I can generally handle it on my own. I try not to worry too much and constantly reminds myself that this is just the beginning of my artistic coding practice, as I recently completed my first semester. Nonetheless, I still do believe that I am getting a little closer to becoming a creative coder. In actuality, even though I think my projects occasionally lacked interactive creativity, my mind didn't. I had a lot of crazy ideas, but insufficient coding expertise to carry them out. This is what I want to work on during the break. My objective is to become an expert in JavaScript so that I can easily produce projects that resemble my Jam assignments. Moreover, my perspective on coding has evolved as a result of realizing that I must approach it from a computational perspective. This means that I must consider how the computer interprets the information I enter and how it will visualize it. I really like JavaScript as a coding language, because I believe that there is a lot of diversity in using it for creative coding. It can be used as a graphic design tool, integrated into other programming languages like HTML and CSS, and used to create both tiny and large games. In fact, these are the things that I can work on incorporating into my creative coding practice. In the end, I am happy with what I learned, and if this is the improvement I am able to have in only four months, I am impatient to see where my skills will be after my three or four years of university in Computation Arts.

Dec. 4th, 2025 Nerly Cadet