

Jake Hayduk

CART 253 A 2252

Pippin Barr

December 4th, 2025

Post-Reflection

I first started coding around 5 years ago when I started Cégep, and without any prior experience at all, it very quickly became one of my favourite subjects. It was something totally new and exciting because there isn't much interactivity with other forms of basic design. Although confusing at first, I really started to get into the feel of using CSS alongside HTML and enjoyed giving things smooth motion, transitions, and animations using hover and clicks and such. However, just the two were never enough, because if you want full interactivity, you need JavaScript. So, we started using it and I realized that this is how you make stuff actually happen. This is actual programming and how most things function, using 'if this, then that'. It was so awesome and freeing, lending to the fact that I could pretty much make anything happen on my little websites. Then we used PHP and I wanted to kill myself. The only good thing about it was the fact that we were using it to communicate in SQL with a database, which opened more possibilities on our sites to be able to store the actual data that users enter. Having to learn Wordpress was a downside, but it did teach me more about programming. Of course, a lot of this was all very confusing and I was still learning, but I would often take the extra time to learn more about different functions or go out of my way to learn new things just so that I could add them into my projects because it was super interesting. I ended up working a contract job for the government (CRTC) in my final summer and after graduating, which was underwhelming in the sense that we didn't actually do much, but I did learn quite a lot about accessibility when

creating pages, and a lot of that translates to game design as well.

When I started CART 253, I was a little (very) rusty on my JavaScript skills. CSS is like riding a bike; it's instinctual as soon as you start again. JavaScript for me was more like skating; it might take slightly longer to balance out and falling is going to happen maybe once. At first I felt majorly overqualified in the class, but I was very quickly glad to hear that we would be learning something that I was not in fact familiar with. The only JavaScript library I had used previously was JQuery. So much so that I forgot most of the basic JavaScript syntax. Which is maybe not good but it's so much easier to read and write. At any rate, I still found it helpful when you (Pippin) took the time to reiterate and explain the way that the code was working, or did the debugging examples. As soon as I started learning p5.js, I could tell it was a whole other world of complicated. Introducing the aspect of drawing things to the screen and thinking about JavaScript in a game design sense really threw me. Nevertheless, I wanted to learn more because it seemed really interesting and the class was engaging enough. All the little challenges helped build up the confidence to try bigger, more complicated things. I really do think that drawing so many things to the screen is impractical though and takes too many lines. I'd much rather use images to display characters and such, like pixel art, for a lower file size because I know that would be the primary issue with using images. I just found it so hard to make something that looked nice, so I decided to use little pixel art sprites in my final project. An interesting thing I realized was that all GIFs end up being synchronized, even when drawn at different times.

I have a love-hate relationship with debugging because it can be absolutely infuriating when something doesn't work and you just can't figure out the reason why. What I've learnt is that the problem is most likely staring you in the face, or it's still staring you in the face but you haven't realized it yet because you're an idiot. Regardless, it's one of the most satisfying things

once you finally get something working, especially if it sounded like a really complex problem that you could barely wrap your head around at the beginning. That was how I was thinking for the final project. How can I do this? Is it even possible to do this? The most important thing is to just start and not think too much about how you're going to get there, because as long as it's feasible, no amount of weird math and logic will stop you. It might slow you down though, which was definitely a learnt lesson for me. I'm definitely interested in seeing what p5.js has to offer in the realm of 3D because in the past I have made a cube that you can rotate by dragging with linear interpolation using only CSS 3D transforms and my own JavaScript mathiness. There is in fact a lot more I need to try with p5.js that I just didn't have the time to get to, although this was a great introduction. The library is pretty massive, even just exploring all the audio functions makes me nervous but I want to try them because as a music and audio guy myself, it seems really useful that you can modify the sound so much before it reaches the user, without having to create different sound files for every minutiae.

If I'm being totally honest, I don't know what my future is looking like. How can I really know when I'm not even sure how I got here? The job market is totally different from how it was just 5 years ago, and it will probably change a lot again. Nonetheless, as a creative coder, all I want to know is that I will still have the ability to create. That means that this course and all other resources out there help immensely with determination and having the means to fulfill those creations. I will continue to learn, make projects and imagine new things, whether it becomes a part of my job, or simply a pastime. The idea is to keep improving because it's a great exercise for your brain and can also be very social with the things that you create. In my final week, I spent way too much time trying to get the realtime multiplayer online functions of my

game working, but I loved seeing it in use. Perhaps I'll follow more into that realm and see what I am able to come up with.