

## Post-Reflection

All things considered, my programming knowledge hasn't really changed from what it was at the start of the semester; I already knew how to code in JavaScript, and was familiar enough with the concept of library documentations that I knew where to find information on how to use p5.js. The only change is that I am now familiar with a JavaScript library I hadn't heard from before. That isn't to say that this class taught me nothing, just that this new knowledge lies outside the scope of learning what a few specific lines of code do and how to chain them together to make a program.

As I mentioned at the start of the semester, my prior experience with coding mostly came from programming websites, APIs, and other utility software, so I didn't have many opportunities to work on projects than are more creative than utilitarian. The only real exceptions were the projects I've done in my Unity 2D and 3D classes in CEGEP, which were usually just demos with only 2-3 different scenes (including the title screen). There was also the game jam I participated in, but the group I was a part of used Unity for that as well.

While I found those infinitely more interesting and fun to work on, Unity can feel pretty imposing at times with how complex it is and how much can be done with it. Paired with the fact that it was my only association with creative computation at the time, I didn't have much motivation to make creative programs in my own time. The fact that most of my ideas were for very big, elaborate games certainly didn't help.

In hindsight, I would have much preferred to have this class as an introduction to programming; The basics were very well defined in the learning materials, and they at least served as a good refresher on the topic. The projects given were also much more in line with the goals I wanted to achieve by learning how to program, which is making games. Now that I know about the p5.js library, I feel much more comfortable with doing smaller projects/prototypes, and actually getting started on them.

I think one of the biggest factors for this might be that, compared to Unity, everything happens in one place. In Unity, you keep going back and forth between the project editor window and the IDE where coding takes place: setting your scenes, coding new scripts, adding those scripts to new

game objects, adjusting the visuals of the scene, and so on. Even if multiple files are also generally needed with JavaScript, it feels like there is much less back-and-forth going on. Working on a creative project in this framework definitely feels less imposing.

I also feel less pressured into making nice, polished visuals from the get-go when working in JavaScript, given that p5.js has functions for drawing basic geometric shapes and it's easy to keep track of the variables for it.. However, this might be due to the fact that all of the projects I've made with p5.js so far are relatively small and simple. It would be a good challenge to start applying this to bigger projects, eventually working my way up to doing this in Unity (or any other complex game-making platform) as well. It would certainly help in reaching my clown game dream goal.

All in all, I do feel better equipped as a creative programmer having done this course than when I finished my CEGEP program. There are some projects ideas I had for assignments that I didn't end up working on, namely those regarding the 1983 game Jetpac, that I feel pretty motivated to make outside of class. I'm hoping this will kickstart the process of actually making prototypes for my big clown game.

While I'm still not quite sure in what direction to take this Jetpac inspiration to make it distinct and truly mine, I'm sure it'll come to me while I try to re-create the original. Adjusting the screen-wrapping to be transitions to new maps – kind of how a metroidvania usually does – definitely feels like a good first step, using a list of lists to keep track of the different rooms like I did with the tiles of the mazes in the Variation Jam. Depending on whether or not I end up using something other than a tile-based approach for the rooms' layouts, I might still have to puzzle out how to draw each room properly, but that's the fun part of programming!

One of my key takeaways from this class is definitely to keep my options open in terms of frameworks I'll be working in. I understand how important it is to find the ones that feel motivating to use, and not just that seem like they're better suited for the job. Starting small and having fun with it is the best first step, and its easier to add complexity later on when you have a good base for it. I also got a better appreciation for making prototypes and just trying random things out from this class, which is an important aspect of creative computation.

Even though the current state of the game design industry is, to say the least, in a bit of disarray and rather unstable, it is still the direction I want to pursue. It is what made me turn to programming in the first place, and the most fun I got out of learning the few coding languages I know. Despite how demotivating hearing anything about the latest news in the game industry is, I'm looking forward to actually putting up games on a platform like itch.io, and just generally adding more things to my portfolio.