Post-reflection.

By Emile Bedard

Work presented to Pippin Barr

CART 253, Creative Computation I, Section A

Computation Arts, Concordia University - December 1st, 2025

Reflective essay: thinking in variables.

I started the term with no prior experience in coding. I was familiar with Arduino projects and some computational tools that were using code to execute their systems but always found them daunting and intimidating to understand and never dared to try. The interface of it all was the scariest for me, mono spaced characters and cluttered workspaces seemed professional and only for top tier hackers. At the beginning of the course, even basic ideas such as variables or if statements felt abstract to me; now they feel like intuitive tools I can rely on. Looking back at this seems funny but it is indeed surprising to see how far Iôve come. Learning p5 and JavaScript in this course simultaneously as learning html and CSS in my network culture course was the kickstart I needed to spark my curiosity and start to build a mental model around coding.

I always had a systemic way of thinking, a bit like a tree structure. I explore ideas or problem solving in very cartesian ways, starting with hypothesis and branching into possible outcomes after. This is most often useful to me; it lets me understand new technical jargon and assimilate systemic approaches quickly. In other more organic situations like interpersonal or sociopolitical thinking, this habit overwhelms me. I burn out thinking about every outcome or possible interpretation and often forget how to let go.

I knew this mental model of mine but never doubted how similar it was to code and computational thinking. While getting to know course material and learning notions throughout the course, I became increasingly familiar with the vocabulary and techniques. To be completely honest, I started to change my perception of data around me, I saw birthdays as variables that could be used in a program, I saw social habits as functions and GPS locations as constants for instance Everything stayed familiar, but as with every new skill I learn, I began seeing new applications everywhere. I questioned how they would translate and take shape in this new language of mine.

Considering I started with zero knowledge, I am very proud of what my abilities are now. I can write JavaScript simple functions and variables easily, I can translate systems, logic gates and pseudocode in lines of commands. I use if statements for logic, event handlers and JavaScript objects. But most importantly, I can understand what I write and see where I do make errors. I know how to plot the logic of my data, debug it when I need to and predict outputs according to my code. Most of this knowledge is still in learning progress and is not perfect, still, it is way more than what I could do at the start of term. I feel confident writing basic syntax in JavaScript but still encounter difficulties for more technical functions or events. For example, I know how arrays and for loops work but I noticed I tend to avoid them most of time because I am scared to write them wrong or to introduce new bugs to my code. I want to learn more about them and get familiar to introduce them in my practice increasingly.

Coding for me is about creating a map, creating a guide of how you can get data from point A to point B with the best route possible. Considering this imagery, I would say that my map now exists! It is very old, from the 1960ôs-ish so some streets donôt exist anymore and Iôm not aware of a lot of new ones. but I can manage to get there, with a lot more wasted time. I know I have a lot to learn, but my foundation is there, I understand how to think about code systems and how to arrange data in the most effective ways. I need to perfect my syntax and learn new notions to get even better, but it is only going to be the cherry on top.

With my new abilities in creative coding, I can expend my mediums and processes. Combining my technical, hands-on abilities I learned from C®gep to these new digital abilities, I can create interactive experiences tailored to my mission. What I mean by that is that by reducing the gap between my mission and what I can physically do, my art is getting stronger.

I did a project a year ago where I wanted to use 3D depth data from a specialized camera to track the rotation of a big plastic eyeball and make it stare at the audience. In this project, I knew how to build the physical components but had no idea how to make it track like I wanted it. Considering my coding learnings now, I think I can figure it out. I could use an external piece of software to plot a point at the center of the detected body and decompose the origin of this point on the X and Y axis and store the values in a JS object. I would then assign and translate using map() this position to the rotation of my two servo motors using live data with serial output to an Arduino and control where the eye is pointing. It feels magical to unlock these new capabilities and consider everything I can now make. I feel like the more versatile and the richer an artistôs practice is, the more freedom he has to convey a certain message. This technical knowledge is also complementary to the experience; it helps bring his ideas magically. For example, I used projection mapping on a tangible acrylic framed canvas last year. This combination of technology, mixing traditional arts and new media made the installation whimsical, people had no idea where the light came from, and the moving painting seemed magical. To achieve this result, I didn’t need coding, but I imagine I could have done way more if I had the tools I know now.

Even if my installations can be better now with JavaScript and coding implementations, I still have a lot of challenges. For instance, In the eye example I pointed previously I mentioned an external software that could plot the point I needed. This would be my limit, if this hypothetical software needs manual intervention or understanding of its processes to use it, I know I wouldn’t be able to make the most of it. In other words, my limit is around basic computation for now and everything that stands out of this spectrum is intimidating. I know this limit will never stop expanding if I work with it and add new learnings to my practice. With any new tool I learn, or habit I try to build, I know the most difficult part is only starting out, building on top after is the confronting and fun part.

Looking forward at the future of my practice, I know I won’t become the most advance coder on the planet. I truly want to expand my experience and learn more about computational thinking to apply it on projects and ambitions. For future projects, I want to explore arrays of objects to manage multiple interactive elements at once, and I want to learn how asynchronous JavaScript can help me manage real-time sensor data. I want to reduce the creative gap and bring back the magic, help people experience unique moments and experiences that are forged forever in their memories. I will continue to learn the basics in school, but I think I will still have interests after. With the help of the online community and resources, I will try to learn new code by my own for specific projects and continue expanding my limit to reach my artistic missions and objectives: Spark strong emotions and unforgettable moments to humans around me.