This being a reflection on my programming knowledge about my courses in computation arts, going over my exploration of programming as a medium seems appropriate. Many of my first projects with programming had a lot to do with web development. When most of my classmates would use Wix or other web development services, my websites would be hardcoded with HTML & CSS. Going further along in school, it became apparent to me the difference between the fields of computer science and web development. Although both share important parallels, web development falls more in the realm of advertisement and marketing. The focus was put on SEO and how to guide the viewer through a page to lead to a sale. The fundamentals of the box model and responsive design were taught but were not in the foreground of the teachings. Before the start of the program, I had the opportunity to spend some time with Concordia's communications department. Being alongside creatives of different fields, the workflow between the photography team, the design team and the web team made for an insightful internship before the start of my first semester of university.

With this pretext in mind, what should I make of the contents of my computation arts classes in relation to what I might want to achieve in my creative programming practice? At this point in time, I have understood some basics of object-oriented programming. My first attempt at making a game with OOP last semester gave me a semi-good understanding of how different blocks of code interconnect with each other. Having a parent class and using keywords that extend into subclasses made for easier implementation of certain parts of the games I would try to make. I also often found myself getting better at prototyping when thinking of a project through this framework. In the previous semester, I would spend a lot of time trying to

hardcode some of the details of the game without having the bigger picture in mind. At a certain point, I had concluded that the libraries we were working with, whether it be P5 or Phaser, were not built for very accurate renderings. It was almost working against me in some of the jams this semester. P5 abstracts many of the more complicated aspects of web development to leave room for creative expression. Going further into the complexities of JavaScript would only put me backward in trying to achieve the end goal of an engaging and creative exploration of a game.

The bias of wanting to work on content first is not new for me. It gives me a chance to catch my breath and have fun making sprite sheets and pixel art while still progressing on my project. Although the basics of programming are starting to feel familiar, something as simple as parameters for example still bugs me. Because of the nature of this program, I can get away with these deficits in my computer science knowledge. I find that I know just enough to bring projects to completion. In the next few semesters, I want to focus on getting my coding skills to a point where sketching an idea is not hard. The closer I can bring the gap in skill and creativity the better my projects will turn out to be. The contents of my 300-level courses in the next few semesters will be the catalyst for how my creative practice will evolve. I will have a better idea of how to materialize some of my creative ideas. Understanding how to leverage my media theory classes and bringing some of those themes into my exploration I think will make for more creative projects, especially if I put the effort when engaging with the readings. With that in mind, I also intend to take some COMP classes outside of CART. I am currently enrolled in COMP 218 – Fundamentals of Programming. According to the outline, it seems to tackle more

computer science concepts through C++, a language I've been meaning to get familiar with. The last piece to orienting my programming experience is getting myself to do more research through scholarly articles.  In my CART 210 class, it was the first time I reviewed articles about game design. It made me want to do academic research instead of following trends through algorithms on my feed. When listening to panellists during the FFAR lectures, each artist spoke with a greater understanding of how their work would help shape the landscape of their respective practices. Gaining a better understanding of how to direct my creative practice I think will help me guide my programming knowledge. Hopefully, with some better time management, I'll be able to pay a visit to TAG–Milieux and gain some perspective on how research can help with my practice.

To conclude, the exploration of programming within the context of computation arts courses has provided valuable insights into the intersection of creativity and technology. Beginning with a foundation in web development, the distinction between computer science and web development became clear to me, with a focus on creative expression rather than a focus on marketing. Getting a better understanding of object-oriented programming (OOP) facilitated a deeper understanding of code structure and project prototyping, improving efficiency in all my projects. When understanding the limitations of certain libraries, this allowed for a balance between skill development and creative expression. Looking ahead, the upcoming 300-level courses offer opportunities for further integration of media theory, enhancing creative projects with conceptual depth. Additionally, enrolling in computer science courses outside of computation arts, such as COMP 218 – Fundamentals of Programming, will broaden the

technical skill set, particularly in languages like C++. Embracing scholarly research, inspired by experiences in CART 210 and FFAR lectures, will hopefully enrich my creative practice by grounding it in theoretical discourse.