Entering this course and program, I have a moderate level of programming experience. As a child, my parents gave me a book about Python for beginners, which was already outdated at that time. I copied each line from the examples to make a simple text-based number game, understanding nothing about the syntax or the logic behind it. Still, this gave me the inspiration to build things from scratch, albeit without the motivation or knowledge to do so. Throughout the years leading up to high school, I experimented with making games with various engines, but I never felt very connected to game design. I took computer science classes from Grade 10 through Grade 12, where I made a few games but found more value in learning about the creative side of programming. My favourite part of making games for class was designing assets for them, like the character sprites in my Tiny Tower clone and the level design in a Stardew Valley-inspired fishing game. Outside of school, I've made a few websites for various purposes, but most recently I was commissioned to build one for a local event planning company. I was given a lot of creative liberty for that project, so I'm quite proud of how it turned out. Programming as an art is one of the most underestimated mediums in my opinion. There is so much that code and computing can unlock across disciplines, like procedural and interactive media, as well as applications in music and even theatre and performance.

I visited New York City in 2023 and decided to check out the Museum of Modern Art, and the first piece being showcased when I walked in was a 24-foot square media wall displaying a generative animation that looked like a liquid swirling around a container like a lava lamp. I later found out that this is one part of a group of works by Refik Anadol titled "Unsupervised". The one displayed prominently at MoMA was trained on the museum's catalogue of art and uses the movement and sound of the room it is in as input. The project as a whole has connections to blockchain and NFTs, which I personally do not feel connected to, but I feel inspired by the art nonetheless. The programming behind it is unlikely to be particularly complicated, but the logical process of continuously generating a video in real time is something that I would be very interested in learning more about. In a similar fashion, another example of creative programming that I enjoy is projection mapping in exhibits. One of the reasons I wanted to explore Computation Arts was from a video I saw of someone who used an Xbox Kinect to track visitors' motion to distort an image that was projected in 3D space in front of them, giving the illusion of physical interaction when nothing was moving. In contrast to the previous example, this is a more mathematically involved task, as simulating realistic, non-abstract 3D imagery in an efficient manner using code is no simple feat. The third instance of programming in a creative way that I find exciting is the way that game developers can express a feeling in a game through minor changes in code. For example, Astro's Playroom is a game that tunes its physics engine along with its graphics and gameplay in a manner that amplifies the

overall feeling of the environment. The character feels bouncy yet robotic, due in no small part to the well-programmed interaction design that also uses the controller's haptics to subtly aid the player's immersion into the world. Contrast this with another game that was released in the same time period, The Last of Us Part II, a story-driven action-adventure game that also uses its physics to drive a narrative, however through purposefully limiting means. The player moves slowly, and like Astro's Playroom, gravity and kinetics match the realistic graphics. A more adult game like this obviously will not use playful elements, but these slight changes in programming make the experience of gaming feel more refined.

The term "generative art" has changed since I was first introduced to it. A few years ago, computer-generated abstract artwork was all over my Instagram feed, with studios collaborating with companies like Microsoft to create beautiful images without photography or hand-drawn art. Now, "generative" refers to something quite different. I would now classify the pre-GenAI CGI as procedural or algorithmic art. Creating art from code autonomously is something that I would really like to work my way into doing. I want to create something interactive, like the project with the Kinect, abstract like "Unsupervised," and dynamic like Astro's Playroom. The first idea that comes to mind is some sort of surreal object that uses an accelerometer and gyroscope together with some physics programming to simulate particles moving around inside it. Alternatively, a large-scale display with some sort of procedurally generated graphic using its surroundings as input is the kind of project that I would like to create eventually.

In conclusion, my journey with programming has evolved from a childhood curiosity to a deeper appreciation for its artistic and creative potential. Through various experiences, I've come to realize that code is a medium for creative expression. I've gathered lots of inspiration, whether from art exhibits, interactive installations, or just simple design choices in video games, which have shaped my desire to explore the intersection of technology and creativity. I hope to further develop my skills in computational art, where I can combine my passion for design with programming to create immersive and dynamic experiences.