Diana Edvi

Professor Pippin Barr

CART253B: Creative Computation

September 9, 2024

Pre-Reflective Essay

I remember the very first time I made a sprite move in Unity. I was following a tutorial on YouTube, and I got so excited about using code to make my character move that I showed all my family and friends what I had achieved. It made me feel so powerful, and it motivated me to learn more about coding. I am currently starting my second year in Computer Science and have already come across many different programming projects that inspire me to continue honing my skills. Here, I will briefly talk about my programming experience and programming's connection to creative expression, as well as some cool projects that inspire me.

There are so many different types of languages, each with their own strengths and weaknesses, that the possibilities of types of projects that can be made are endless. I am still at the beginning of my programming journey, but I have picked up some skills over the last two years that have given me a solid foundation in programming. I have taken Object-Oriented Programming I & II, and I also began learning a Udemy course on web development over the summer, which has so far taught me basic HTML and CSS. I have also participated in a few game jams wherein I tried making games with C# in Unity.

One of the first lessons I learned after beginning my programming journey is that the range of what is possible is only limited by your own capabilities. I have a mentor friend who has been working as a programmer for about 20 years, and he told me, "Everything you want to do is possible. You just don't know how to do it yet." It is therefore not hard to see how programming is a great means of creative expression. With the right language, and enough determination, you can create any program or application that you can think of. From games to web apps to AI, the range of what can be done is practically endless, which then gives people unlimited opportunities to utilize their skills in unique ways. Also, programmers have complete control over how their program will work and be presented. They are not bound by templates or pre-existing structures, so they are able to present their ideas exactly how they want to.

Over the last few years, I have come across many inspirational coding projects that made me want to improve my own skills, mostly in the form of YouTube videos. For example, I watched a video essay about a game called Rain World, wherein a "slug cat" tries to survive in a harsh environment. The game's uniqueness comes from its intricate AI system, which simulates a realistic ecosystem. Every creature in the game is its own entity with its own list of needs and

wants. The player is meant to be treated as just another creature in the world, and any action the player does has a butterfly effect that travels throughout the entire ecosystem. I found this aspect of Rain World to be truly inspiring because I had never seen AI used in an indie game to that extent and I was very intrigued by how everything worked.

Another source of inspiration for me is the community of people on YouTube who are determined to build computers in places where computers should not exist. For example, building Minecraft inside of Minecraft, building a CPU in Excel, building a computer that runs other games in Terraria, etc. One of my favourite courses so far in Computer Science was System Hardware, and I found it incredibly rewarding to watch these videos and (mostly) understand the intricate low-level programming that was required to complete these projects. It made me want to turn around and find a project to build a computer in, although the most I've managed so far is playing with Redstone in Minecraft.

The last project I wanted to talk about was a handful of videos made by a small creator called Zyger. I am inspired by her because she is around my age and has already accomplished so much. Her journey is a sign to me that I am on the right track. Of all her videos, the ones that inspired me the most are the ones where she takes a crack at building her own engines (physics, graphics, etc.). When I first saw these videos, I was completely blown away. It didn't even cross my mind that I could even do something like that. The idea of building an engine completely from scratch, starting from a blank window to having objects rendered and be able to be manipulated on screen, was a complete game changer for me. It took a hold of me so strongly that it gave me my idea for my "dream project."

I have recently set a personal goal for myself, which is to try and make a simple game engine from scratch by the end of my degree. It wouldn't be anything complicated; it would probably just be a bunch of cubes you can manipulate, but it would still be cool to do. This project would test my math skills, my ability to write good C++, and probably my patience. I really want to do this project because I am intrigued by how things work. I want to be able to understand how a massive engine like Unity functions on even just a basic level. Also, learning C++ is something I have always wanted to do because it is apparently a very strong asset to have in the game industry.

Overall, my experience in programming has been largely shaped by the courses I have taken and the projects that inspired me. Switching into Computer Science was one of the best decisions I made, and even though it is challenging, I absolutely love learning new things and honing my skills. That is why I am very excited to take CART 253 as well. JavaScript seems like a very useful language to learn, and I am excited to see what I can do.

Here are some of the videos I mentioned:

Rain World: https://www.youtube.com/watch?v=6Ji2q3WQE78

Terraria computer: https://www.youtube.com/watch?v=zXPiqk0-zDY

CPU in Excel: https://www.youtube.com/watch?v=5rg7xvTJ8SU

Graphics Engine by Zyger: https://www.youtube.com/watch?v=XfKFZ46mDXI