Pre-Reflection

Having completed a 3-year program in computer science back when I was in Cegep, I have a pretty good deal of experience when it comes programming in general. During that time, I learned to code applications with various languages, mostly c#, python, html/css/javascript, java, and Kotlin, as well as applying concepts such as good design principles and object-oriented programming. The most fun I've had in this program was while taking the Unity 2D and 3D classes, since those assignments were more creatively inclined than the other classes. In the former, I even independently learned rudimentary 2D animation skills so I could use my own assets for one of the projects. Because of that, I've been mostly gravitating towards Unity games as my programming of predilection for my personal projects. I'm very excited for this class since I'll be able to use my pre-existing knowledge to make interactive projects in a more web-based environment, which I haven't really used for this kind of work before. So long as you can figure out the logic for what you want to make and the framework needed for it, you can do practically anything through programming.

Most of my creative inspiration comes from video games, obviously. One of the most recent ones is *Oneshot*, by *Future Cat LLC*. What really amazes me about it is how much they're able to do with the RPG Maker engine, and that they've coded it to use your whole computer as part of its interactive design (at least, for the original Steam version). To be able to advance the story, you need to look around in your computer folders for specific files that the game gradually adds, not always in the same place either, and then interact with those in different ways. The only other time I've seen a game use that concept is in *Doki Doki Literature Club*, but even then it was pretty much only used at the end of the game. There's also a specific moment in *Oneshot* where the game has you run a second executable to help you progress by showing different things in that second window, depending on which room you are in in the actual game window. The programming behind this must be really interesting, especially with the RPG Maker engine. I'm definitely interested to learn of more projects that use similar mechanics in their design.

Another game that I find to be a feat of programming, although in a completely different sense, is Toby Fox's *Undertale*. The entire game is practically run from within some hundreds-lines-long switch-case statement, and that barely an exaggeration. As a programmer, this is painful. As an

artist, this is amazing. This game has made it into the hands of the Pope and it's among some of the worst plates of spaghetti ever coded that I know of. It just goes to show that you don't necessarily need incredible knowledge of how coding works to make something amazing.

Last but not least are randomizers in general; Fan-made programs that let you create a unique configuration of a game every time you press a button, with extremely customizable settings in some cases. I even know of a co-op randomizer where the players will individually play entirely different games, and actions done in one game can unlock important items to progress in the other game and vice-versa. I can't even begin to imagine the amount of (completely voluntary!) work that is put into these programs to make sure that, for most settings at least, any given "seed" of a game can be completed, or even just how they're able to take an existing game, like *Donkey Kong 64*, and shuffle everything around in the first place.

As for my creative programming dream, it is absolutely my Action-Adventure RPG Clown Game project that I've been slowly but steadily designing for the past 2 years or so. I haven't started with the programming yet as I want a better grasp on the story I want to tell and the characters that will take part in it, but it's definitely going to be a part of the process that I'll enjoy a lot. When I'm coding a game project or prototype, I like to see it as a sort of jigsaw puzzle; I need to first find the pieces that make up a specific game mechanic I've come up with, make them fit with each other, then once I have a few of them going I figure out how they all fit together for the big picture. Figuring out how these mechanics usually work and being able to use them to make my own games is the reason why I wanted to learn how to program in the first place, so I'm looking forward to this part of the game-making process. In the meantime though, I'm having fun designing and drawing some silly little clowns, and thinking about the worldbuilding.