Émile Jolicoeur 01/31/2024

Creative Computation II:

“Preflective” Essay

I was first introduced to the world of programming over five years ago, while I was in a course elective focusing on the conception and creation of videogames. I had wished to make a program of my own, as the function I required for a final project was unavailable to me. It was a tremendous task for someone who had no previous knowledge of programming at the time, but after four days, I had finally managed to figure out the 4 lines of code I needed. This was how it all started.

Since I was twelve years old, I have been set on having a future in the gaming industry. I was always fascinated by the visual aspects in the past, but recently, while learning the basics of programming throughout the previous semester, I have gained a growing interest on learning the inner workings of video games. I managed to create a somewhat complex game on my own using all the knowledge I got from *Creative Computation I (*struggling a bit with loops and arrays as I did) heavily based on the game *Keep Talking and Nobody Explodes*, and finally saw my potential for programming.

Considering I am aiming to one day be a part of the gaming industry, it shouldn’t be surprising that most of my inspiration comes from video games, programming being no exception to the rule. I am amazed by the genius programming which lies behind games like *Doki Doki Literature Club* (2017), or Valve’s *Portal* duology. In *Doki Doki Literature Club* (or DDLC), the player could search the game files (or at least fake ones) to uncover secret images, sound files and messages which would appear after reaching certain thresholds in the narrative. Interacting or even deleting those fake game files such as the characters would remove them from the still functional game. This kind of program, while not seen by the player, adds depth to the game, making it feel alive and sentient! Another feat of programming ingenuity which still has me in awe to this day is the creation of physical, working portals in *Portal* and *Portal 2*. While they are visually mind-blowing in the game, I can only assume that displaying a real-time image onto a surface is child’s play compared to the massive script concerning the character’s physical properties (speed, angle, hitbox, etc.) and essentially teleporting it from one side to the other with such smoothness. This aspect who’s conception remains inconceivable to this day, is an amazing feature I wish videogames used more often.

If I am to create complex video games in the future which will have a similar impact as Portal and DDLC, some of the more important elements I need to learn would be how to save and store variables for later use, which won’t reset after closing the game. This seems to be an asset I might learn in the optional documents available on week 3. Another useful piece of knowledge would be the use of AI tracking, especially as a tool for animating a character. This knowledge will be seen throughout the fifth and sixth week of the semester. Finally, it goes without saying that knowing how to use a game engine would be of great help in order to reach my future in the gaming industry. Fortunately, this too will be viewed this semester, right after the reading week.