Viviana Ema Radu
Preflective Essay
CART 263

- Discuss aspects of your existing programming experience that you find exciting in connection with creative expression. This can include elements that you are comfortable with, but also things you're challenged by but want to incorporate into your creative coding practice.

I studied computer science throughout my CEGEP degree, so I'm actually very familiar with programming and have done work in C, C++, C#, Java and Javascript. I'm particularly familiar with working in video game engines such as Unity and Unreal Engine, and have dabbled in RPGMaker MV. My direction in programming has always been oriented towards video games; I play a lot of them and I love making them. I'm particularly interested in smaller, more conceptual games that offer an experience or evoke a certain feeling in the player rather than the traditional linear cohesive form of games. I'm getting more and more familiar with using these game engines, and while they are very comfortable to use and offer a lot of interesting features to build games that I envision, I find one aspect I tend to struggle with is the artistic direction of my games. When you rely too hard on engines with pre-coded aspects such as shaders and UI features, it's hard to even get started on learning how to code my own and imagine a program or visual software that has a specific look that I want to give it. When I step out of these engines' limitations which create clear boundaries for me and try doing something by myself, it becomes a lot harder for me to have a clear grasp on where my artistic direction should go and what my game should look like.

- Present one or more examples of creative code/software you are particularly inspired by and why. Note that the inspiration here should be drawn from the programming aspects rather than narrative, audiovisual aesthetics, game design, etc.

I think one thing I would definitely like to try doing more is dabble in games that start simple and expand largely on that base. I think I have a habit of starting with major game elements that are already very complicated, and on which expanding will mean having to add onto all the other major elements, and so on. This means that just finishing the bare minimum of gameplay takes a lot, and I give up on expanding it further. I want to start with simple concepts that can easily branch off into different things so that my games become more dynamic to play the more they go on. It also gives the opportunity to randomly generate smaller game elements rather than have one big thing set in stone.
Recently, I replayed the game “Papers, Please” by Lucas Pope. The game has simple gameplay: you play as border patrol, you check randomly generated documents from fictional people which could have discrepancies, you may question them, and either grant or deny them access. Every day introduces a new feature you need to keep track of on top of the pre-existing ones from days prior, sometimes replacing them, and it becomes increasingly hard to remember everything, especially within the time limit of each day. Most of the game is text-based and has randomly generated elements. I

really enjoy how his games have simple gameplay that still becomes excessively challenging since it relies on your memory and, most of all, mastering all of the game's concepts even as they are constantly changing.
Another game I absolutely love is Daniel Mullins' "Inscryption". At its base, while being a story-driven game with a lot of deep lore, most of the game is spent playing a card game. The card game in "Inscryption" starts off very simple, with a three-card deck you expand on the more you play, as well as randomly generated card battles and events to power up your cards. Since it works off runs and each run is randomly generated, your strategy can change drastically depending on the cards you're given and on the maps that are generated. It gives off a roguelike feeling in this aspect, while at its core it's a very simple card game, and you always start with the same cards! But a concept like this can start simple and be expanded on. Most of all, gameplay can alter greatly with each new card introduced, since it changes the synergy of every card before it, without the need to re-code anything: the gameplay itself hasn't changed, but the player changes how they play it.

- Describe one or more big ideas you're interesting in exploring through creative programming with some initial thoughts on what you'll need to focus on to make them happen; make connections to the course outline and schedule to link these ideas to the course itself.

I'm actually very curious about AI and creating more complex AI within the context of video games. We will be learning AI during this course so I can't wait to see what I will be able to to with it. I think AI being inherently interactive adds a lot of potential to a game, since it learns to behave specifically to react to a player or user. Games often use AI as an "enemy", or to go against the player: enemies in battle-based games have an AI enabling them to attack and react to the player's attacks, competitive online games like chess have an AI on how to counter a player's plays. AI often replaces a real enemy player, and even in multiplayer games, AI is put in place for practice sessions or tutorials to emulate what playing against a real person would be like. I think what tends to be janky in AI is when the AI becomes your ally. AI never quite acts as clearly in your favor as when you play with a real person, since you can discuss and form a strategy together, and games often don't really offer many ways to communicate with those AI in depth either. I think an AI you can communicate with under a cooperative context and who mainly is built to understand and respond to communication and each individual person's play style with its own actions which can lead you both to victory would be a great way to explore the relationship we have with AI in games presented as something we can form a connection with, rather than something we have to trick or outsmart.