Concordia University

Faculty of Fine Arts - Computation Arts

Creative Computation I

CART-253

Project 2 - Proposal

Presented to:

Pippin Barr

By:

Nicolas Morales-Sanabria

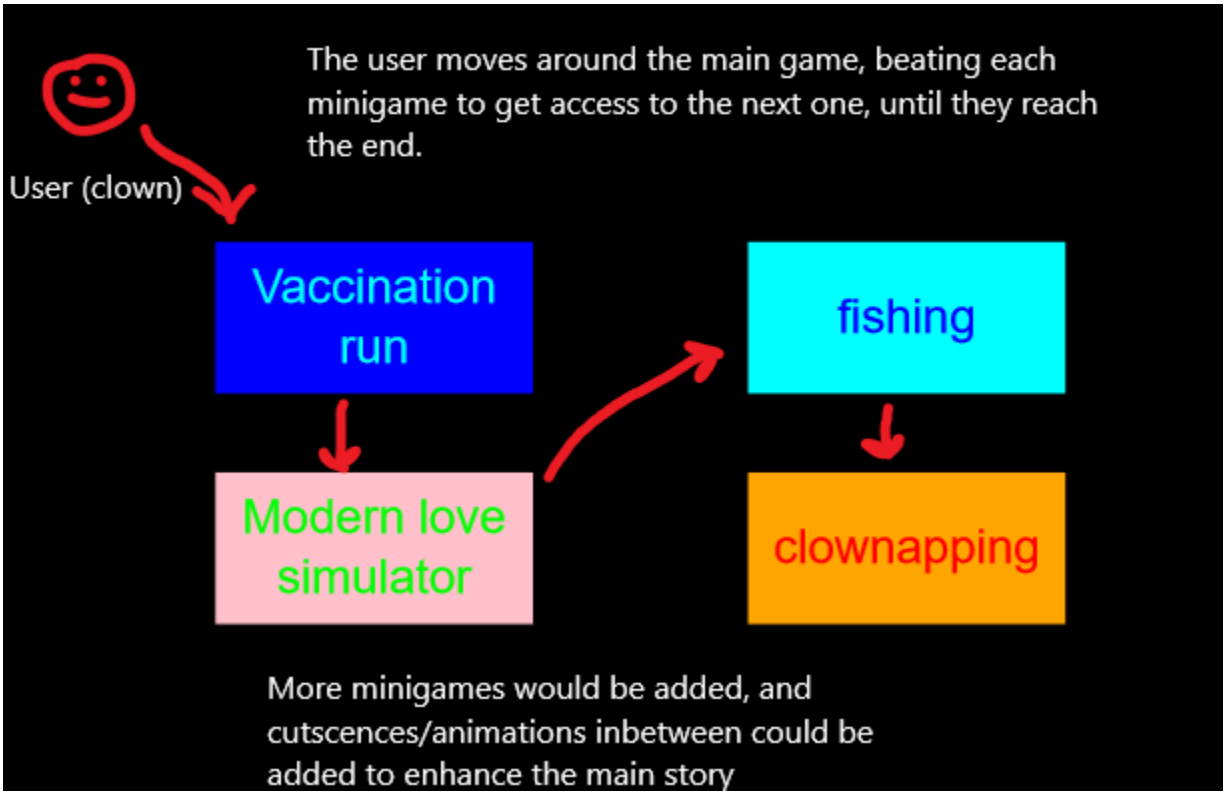
November 7th, 2023

The artistic vision for my project 2 is to make a minigame library that contains multiple games, the twist is that the library will be a game itself (and will use its minigames to tell a story) where the user will be able to access new areas & games as they progress. The minigames games will all be connected through the theme/story of the main game, launching the "minigames" as events happen in the "main" game. The prototype was developed to build a project that incorporates & runs multiple games in the same script, offering a solid base to build on, it also explores the use of OOP in this context & avoiding duplicate code through the creation of a library. I believe this concept will allow me to express my style & ideas as a programmer through the different games, their styles and mechanics. Although I believe having games inside a bigger game is an interesting concept to explore, it presents a few technical challenges:

1. Object-Oriented programming: OOP is needed to make the code manageable & having multiple games will require an organized class/file structure to make the project clear and manageable. My prototype explores this concept with a folder structure and classes for games & objects, aiming to keep everything clear and organized.
2. Sound: I plan to find and add sounds to the games, which is something none of the current prototype games have. I'm not worried about it as I just have to learn how to import/play 1 sound and then I can repeat the process & progressively add sounds as I advance on the project.
3. Main game/minigame communication: I believe the prototype has a solid base for how the minigames should independently run, however it doesn't explore how the minigames will interact with the main game (menus for the prototype, to be changed). I would approach this by using global variables to communicate between classes when certain conditions are met.
4. Avoiding duplicate code: The multiple games in the prototype would sometimes have functions shared by multiple classes, to avoid having duplicate code I explored creating a p5 library (p5.commonGameFunctions.js) where I placed general functions that could be used across games, it helps in avoiding duplicate code by centralizing common functionalities. I see two principal benefits to creating this library: Firstly, I was able make the classes for each individual game more concise and only leave functions more specific to each game, secondly the functions in the library should make it easier to create more games as I will already have those useful functions. Another positive of making this library is that it can later be shared to more people.
5. Making sure the project is "finished" by the due date: Because this is a collection of games, and I could in theory add as many games as I can and indefinitely extend the main game. For this reason, I need to make sure that the main game is not too complex and whenever I add a new minigame, it is integrated correctly. I will take precautions towards the end of the project to prioritize the program running correctly instead of trying to add unfinished minigames that might generate errors/crashes.

Sketches:

Structure of the main game (example, not in final order, to visualize the idea of progression)



Sketch for the main game (very basic idea, not final, to represent how the progression could be like):