Andrew: johnliu
Name: John Liu

TP 2 UPDATE:

Multiple inputs have been resolved and implemented before MVP. Goals such as 3D visuals, audio, mp3 importing, and openCV will be implemented after TP2 meeting if MVP is confirmed.

TP 1 UPDATE:

So far, I have noticed that it is difficult to take multiple inputs using cmu graphics (using keyboard inputs before MVP). I was planning on using multiple inputs simultaneously but I might have to reduce it to one input. Other than that, I don't find a real need to change any other parts of my project.

Project Description [2.5 pts]: The name of the term project and a short description of what it will be.

The name of my term project is "Dancerush 112." It is based on the rhythm arcade game "Dancerush Stardom." The overall project will be based on the premise that there will be rhythmic notes and sliders coming down on the screen and the player is meant to create user input in the game to match the game objects. The player input will be created with OpenCV which will track the player's feet movements to mimic the input of the dance board in "Dancerush Stardom."

Similar projects [2.5 pts]: A 1-2 paragraph analysis of similar projects you've seen online, and how your project will be similar or different to those.

In the past, there was a project called "Beat Beat Revolution" made by Paige Yu. My project will be similar in the user input matching the game elements. I would also like to incorporate an auto generated map element into my project where a user can import any song. The differences I would include is complexity in varying the difficulty of each song and a 3D visual effect to the game elements to recreate the visual effect in DancerushStardom. Another project, "Beat Slicer" by Andrew Zhao incorporates the 3D visual effect and OpenCV components I wish to add.

Structural Plan [2.5 pts]: A structural plan for how the finalized project will be organized in different functions, files and/or classes.

My structural plan will use OOP for an overarching game class that will run all aspects of the game. There will be an overall game class that will communicate the game elements and the player inputs and statistics. There will be a player class that controls the player inputs and score. There will then be a Maps class that generates the map of nodes and sliders (game elements), which will create and store instances of different notes.

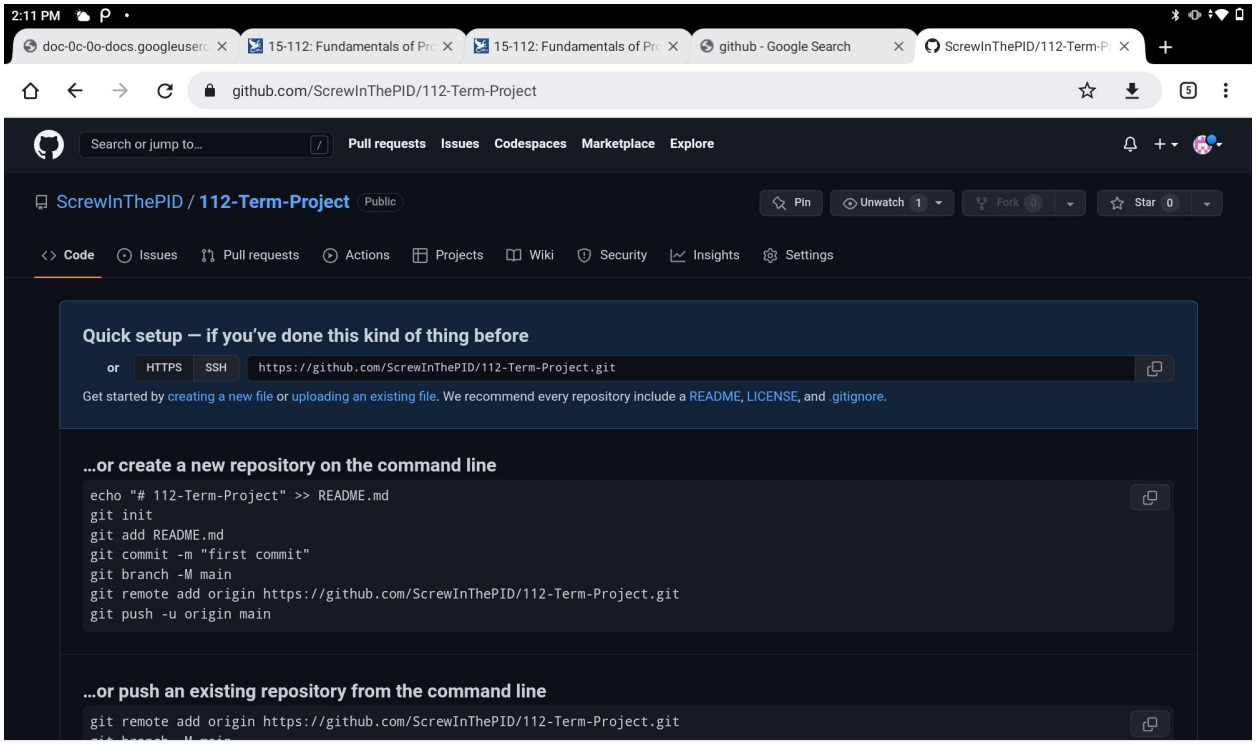
After everything, there will be an overarching display class that displays everything occurring in the game.

Algorithmic Plan [2.5 pts]: A plan for how you will approach the trickiest part of the project. Be sure to clearly highlight which part(s) of your project are algorithmically most difficult, and include some details of how you expect to implement these features.

The main algorithmically complex aspect will come from the OOP structure of my project which I will implement in a series of classes and subclasses as mentioned in the Structural Plan portion. There will be multiple parts to this project that will be tricky including randomly generating the map based on an imported mp3 file and the 3D display that will occur in the main display class. I plan to use the Maps class as mentioned before to implement a detection method for aspects such as bpm of an mp3 file which will play into how the notes are randomly generated along with the random import. My Player class will integrate OpenCV which will replace the keyboard input with visual input after MVP.

Timeline Plan [2.5 pts]: A timeline for when you intend to complete the major features of the project.
Matching user input keyboard inputs to game elements, displaying objects, calculating score and a simple algorithm to randomly generate maps will be completed before MVP. After MVP, I will incorporate the openCV component that will replace the keyboard inputs and then randomly generate maps based on imported mp3.

Version Control Plan [1.5 pts]: A short description and image demonstrating how you are using version control to back up your code. Notes:
You must back up your code somehow!!!
Your backups must not be on your computer (ideally, store them in the cloud)
I will store my code on github using a repository. I will clone the repository on vscode and manage my code from there.



Module List [1 pts]: A list of all external modules/hardware/technologies you are planning to use in your project. Note that any such modules must be approved by a tech demo. If you are not planning to use any additional modules, that's okay, just say so!
I will use a second camera that will track the player's feet for additional hardware. I'll use OpenCV for computer vision and pygame for audio and the aubio library for random map generation.