Kenneth Huang

**Design Proposal**

Project Proposal:

Project Description:

My project, Music Rush, is an audio-based game that generates different maps based on different music files. The game works similar to Subway Surfers and Piano Tiles, and is based off of the Steam game AudioSurf. The main goal of the game is to rack up as many points as you can before the song ends by hitting blocks, while avoiding the hurtful blocks.

Competitive Analysis:

As I browsed the 15-112 gallery for audio-based projects, I found many different forms of the project. In the gallery, there were music transposers, music visualizers, and a couple of music-based games. All of these projects were common in the fact that the projects were personalized based on the player’s music preferences, where players are directed to import a music file in order to get the program to start. I am also planning to implement this mechanic, because the personalization of these projects is what makes them so personal and unique compared to other project topics. However, unlike the rest of the projects, my project will be more gameplay focused, in hopes that anyone who finds my project will be able to have genuine fun while playing the game. By using music and colors, which will be generated based off the music, to immerse the player into the game, and by synchronizing music beats with itself, my game will stand out among other music projects.

Structural Plan:

I am planning to make my audio-based game as user-friendly as possible, with a start screen that leads to a Play and a Tutorial button. If I have enough time, I also plan to make a character selection screen, where the player will be able to choose between different characters with different “abilities”. After the user clicks the Play button, they will be brought to a screen showing the possible mp3 / wav files that they are able to import from their computer. After the user selects their song, the game will generate and begin.

Classes: Enemy (the obstacles), Player (the player)

Subclasses: Obstacles that give points and obstacles that hurt

Another subclass: Different characters with different abilities under Player

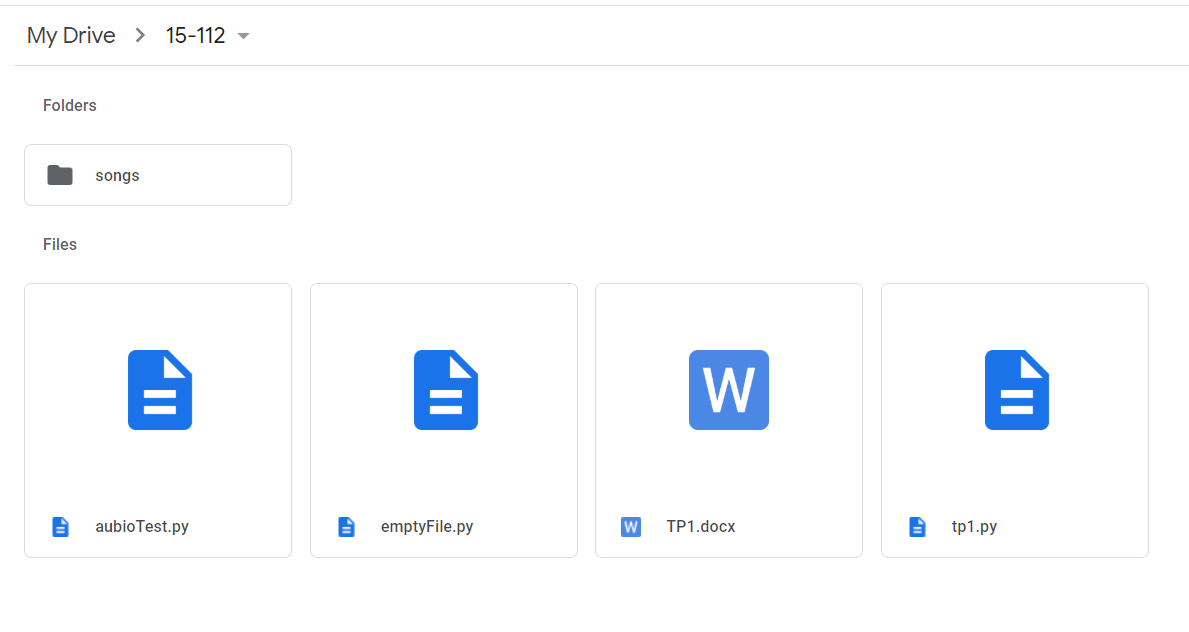
Main functions: drawAll, runGame, and all the different modes

Algorithmic Plan:

The trickiest part of the project is most definitely finding an algorithm to generate obstacles and colors based on the song. For the obstacles, I am planning to generate them using the onset and tempo functions built into aubio, so that the obstacles will collide with the player at the same time a “beat” plays. For the colors, I plan to use brighter colors such as green or light blue for the softer parts of the song, while using red and purple for the more intense parts of the song. I will use the pitch, onset, and tempo functions to achieve this, taking into account the change in speed of the song, the change in pitch of the singer, and spikes in the vocals of the song.

Version Control Plan:

Google Drive



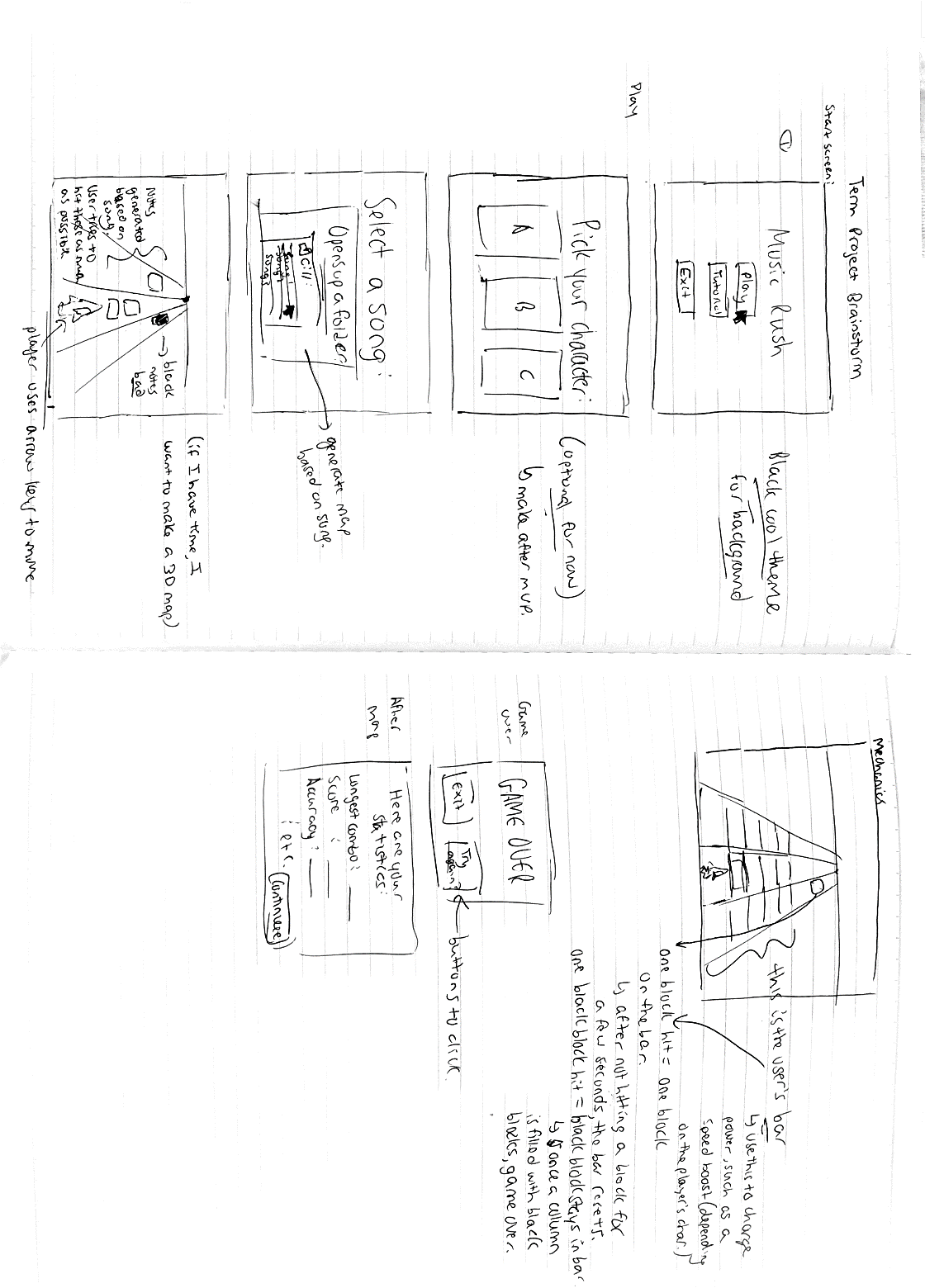
Module List:

Pyaudio: <https://people.csail.mit.edu/hubert/pyaudio/docs/>

Aubio: <https://aubio.org/>

Pygame: <https://www.pygame.org/docs/>

Storyboard:



TP2 Update:

Design Changes: I changed the game from a top view to a one-point perspective. I feel like this will better immerse the player into playing the game. I also implemented an accuracy system, and another rule where the player must click the ‘Space’ bar in order to hit their notes.

Improvements: I successfully found a method to convert sounds into color. Using this, I made a more aesthetic background, and I found this sound-to-color converter to be a very big part of my code.

Problems: I found it extremely difficult for me to create an arc using pygame. I needed to create an arc with a constant center, but the built-in parameters were very unlike those of the pygame circle parameters. The arc parameters required the top left coordinate and the width and the height of the ellipse, so I needed to use heavy math in order to solve this problem. Scaling the obstacles was another problem that came up, since more math was involved and I had to scale the obstacles in relation to its position on the map.

Peer Review Update:

After discussing with my assigned group in lecture, I was suggested to implement a high score function. To do this, I used file managing to create a new text file that keeps track of the high scores of each song. In addition, I added a multiplayer game mode for additional complexity.