Shalini Rao

Section G

Mentor: Katherine Yang

TP1 Design Proposal

Project Description:

My project will be called Dance112. I will be doing a version of the dance game Dance Dance Revolution. Users will be able to use the keys to match randomly generated arrows to a target position. The arrows will be synchronized to a list of uploaded songs, with the possible later capability for the user to upload their own song files.

Competitive Analysis:

I have seen a similar project from previous semesters of 15-112 where the student used a Kinect and users had to match their hands to the corner of the generated arrows. I’m using arrow keys instead because I think it allows for a quicker and more sophisticated gameplay. It also creates the opportunity for harder and faster levels, which would be decreased by using the Kinect to track body motion. This is a very popular online arcade game, with multiple versions using arrow keys to play the game. My project will obviously be coded using Python, which sets it apart from other similar online games. My user interface will also be substantially different from the other versions.

Structural Plan:

Pygamegame: Will access/call all other classes to bring together all components of the game. Will also hold the different screens created by the different modes

Arrow: Class for the randomly generated arrows that move at a given speed - will eventually be configured to change speed based on individual songs

DefaultArrow: Class for the permanent arrows at the top of the screen that determine the user’s score

TempoDetector: will take in a sound file and detect the tempo so the moving arrows can be on tempo with any song

Algorithmic Plan:

I will use PyAudio to track tempo of different sound files of the uploaded songs so the arrows can have different speeds for different songs

I will use:

Aubiotrack - intuitively finds beats in a given sound file - similar to how a human taps their foot to music

Get\_samplerate - returns sampling rate in Hz

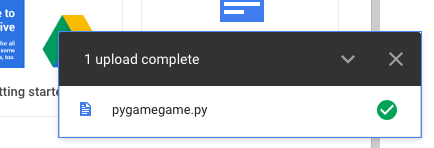
Timeline Plan:

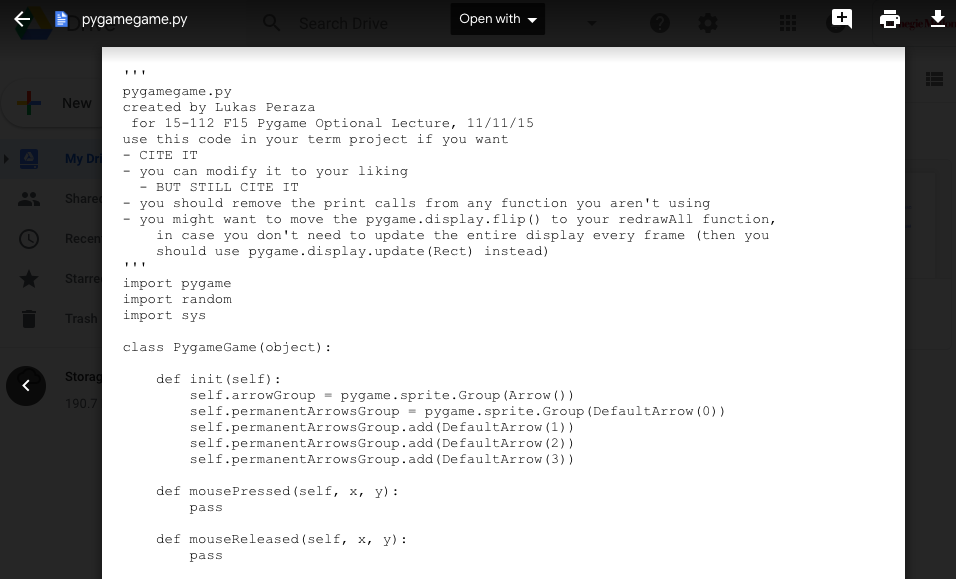
Middle of TP1 and TP2 - be able to play songs at same time as arrows are moving, better interface: score, allot different scores based on when arrow hits the mark

TP2 - tempo detection for different speeds/different songs, all screens created (introduction, instructions, game, ending)

Middle of TP2 and TP3 - adding other things: harder levels - multiply tempo of song, upload own sound files to game, possible multiplayer, more advanced user interface

Version Control Plan:

I will be using Google Drive by uploading my file every time I work on my TP, which saves it online as a text file of my code.



Modules Used:

I will be using PyAudio and Pygame.

**TP2 Design Update:**

I don’t think I will be adding the capabilities of multiplayer or for the user to upload their own songs to the game anymore.

* I will be implementing the beats tracking with a new method:
  + I’m writing a different class that will implement beats tracking to create a text file with the songs and their median BPMs that will be used to coordinate the arrows to the song. When the user chooses their song, the game will find that song’s bpm in the text file and change the speed of the arrows to fit their choice.
* I will be adding the capability of storing high scores for each song so users can compare their scores to those who played before them.
* I will also be adding the capability for users to choose the difficulty they want to play at and changing the speed of the arrows according to their choice.

**TP3 Design Update:**

* I ended up not implementing the capability of changing difficulty level.
* However, I created a function to draw a fractal whenever the user gets a perfectly timed arrow press to match the permanent arrows at the top of the screen