Taiko no Tatsujin (keyboard edition)

A remastering of a classic Japanese arcade game where players hit a drum on the beat of the music.

Similar Projects/Inspiration: Taiko no Tatsujin game, rhythm games in general

Structure

- Scoring System:
    - Score will be based on accuracy of the type of beat as well as timing. Final score will be calculated with a function taking in both of these parameters
    - Will output a score then ranking from D to S rank
    - Combo system to rewards more points for consecutive good drumming
    - Saves best score in a .txt file so that scores can be saved throughout multiple booting up of the application
- Sprites:
    - Iconic Taiko no Tatsujin mascot will have a sprite animation while playing the game to indicate how well the player is doing generally (happy dancing/ sad)
    - Sprite file to iterate through each sprite image.
- Drum types:
    - Each note has a different type of drum the player has to hit, whether it is a 'don' beat or a ‘kat’ beat.
    - Evaluating the following types of beat:
        - 'don': one single hit of the ‘j’ or ‘f’ key
        - Solo 'kat': one single hit of the ‘r’ or ‘u’ key
        - Big 'don': hitting the ‘f’ and ‘j’ key together
        - Big 'kat': hitting the ‘r’ and ‘u’ key together
        - Roll: hitting as many 'don' beats as possible in a row
- Song Selection:
    - Folder with songs and preloaded levels that I create on my own
    - Multiple difficulties for each level
    - Hoping to make 3 levels for term project
- Level Editor:
    - Allows for player to make their own type of level by importing a song, then adding different types of drum beats at respective time stamps
- Two-Player Mode:
    - Allows for two players to play against each other with the same song, with both players using the same keyboard

Algorithmic Process

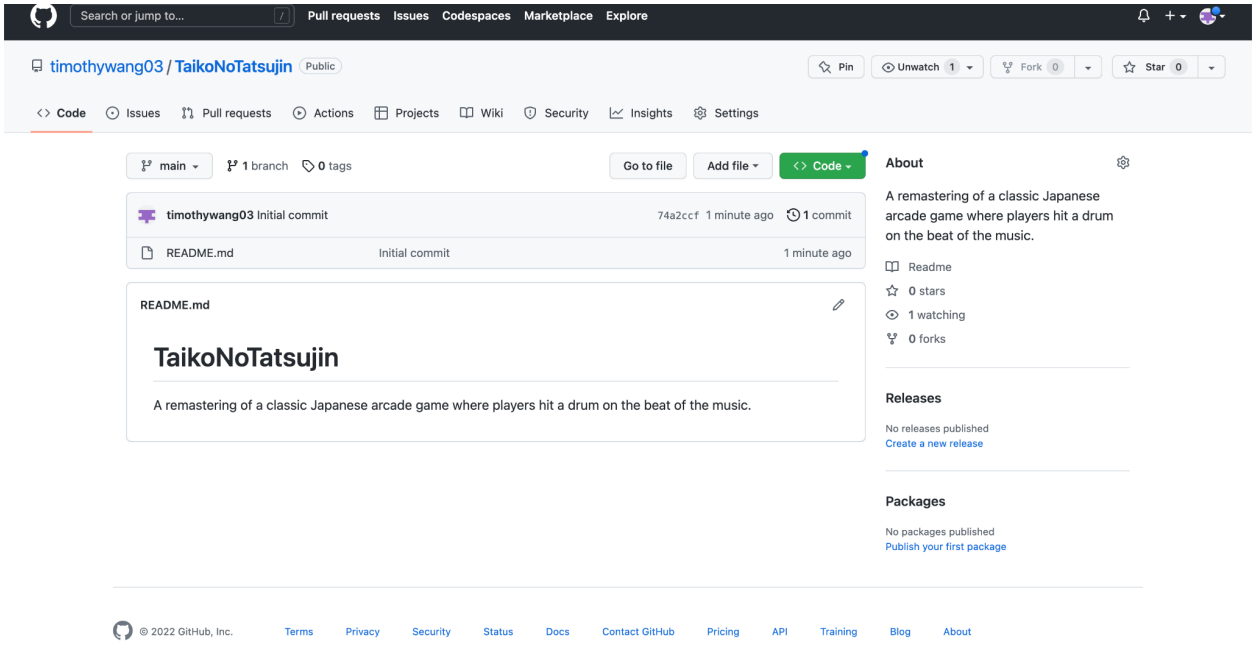
I think the hardest part of the term project will be getting the level editor to work out well. This is because with the user having a very hands on approach, there is a lot of room for them to make mistakes that they are not able to anticipate. In order to complete this, I need to add a lot of error testing. Additionally, I need to hash out the best way to store when notes appear, which I am thinking of using a dictionary with a key as the timestamp and the value as the type of beat. Thus, each type of note will be defined as a class that inherits from the Note superclass.

Timeline

Nov 14 - TP0 Due
Nov 16 - Write Note class
Nov 18 - Write Preliminary scoring function
Nov 20 - Finalize each Note class and start graphics
Nov 20 - TP1 Due
Nov 21 - Finish first level
Nov 24 - Finish level editor, finish all levels
Nov 30 - Start 2 player mode and finalized sprite sheets
Nov 30 - TP2 Due
Dec 1 - Finish Debugging, if time allows start hardware implementation if possible
Dec 5 - Implement drum in substitute of key presses
Dec 7 - TP3 Due

Version Control

Entire project procedure will be tracked using Github:
[github:https://github.com/timothywang03/TaikoNoTatsujin](https://github.com/timothywang03/TaikoNoTatsujin)



Module List

pygame==2.1.2 (for audio)
Cmu112graphics (for overall graphical interface)
Time (for timing)
OS (for file selection)

Tp1 update:

No changes

Tp2 update:

A lot of ambition, lots of bugs out of my control, leading to simplification of song, as well as simplification of sprite interaction