15112 TERM PROJECT

Two player fighting/arcade game
using Python's *Tkinter* Library

PROJECT DESCRIPTION

Two players fight on a platform (that can be selected at start) on the same keyboard using WASD/Arrow keys. Players punch and block for 120 seconds. The one player with the highest HP wins!

Twist: The abilities of your fighter depend on the mask you're wearing, which you can only wear for 7 seconds at a time.

COMPETITIVE ANALYSIS

I watched people play Brawlhalla, a similar platform fighter. It’s got a lot of action on the screen at one time. Another similar game inspired me was Smash Brothers. I plan to make my game much simpler and cleaner than these two games--it’s meant to be a fast-paced game you can play in two minutes with a friend.

Taking into consideration the controls, I’ve decided to shorten the game span to two-minutes to maximize intensity and fun. Sharing a keyboard is an awkward experience to extend beyond a few moments. Having to play a game for 10 minutes while sharing the WASD and arrow keys on one laptop can become uncomfortable for the player. To make my short, unique game stand out from it’s longer, flashier counterparts, I’ll pay close attention to the UI/UX post-TP2 and add small details to make a seamless, exciting experience for the players.

STRUCTURAL PLANNING

Players

- getPosition()
- draw()
- move()
- isTouchingOther()
- attack()
- reactToAttack()
- putOnMask()
- returnToNormal()
- onTimerFired()

Masks, levels are also their own objects.

ALGORITHMIC PLAN

The trickiest parts of this project that I can foresee are:

1) Loading the level
2) Animating player battle

My plan to load the levels (which I have already begun in my starter code) takes inspiration from our Tetris assignment. I have a 2D list of True/False values and will loop through it, drawing a rectangle at certain points of the screen based on whether the corresponding label says True.

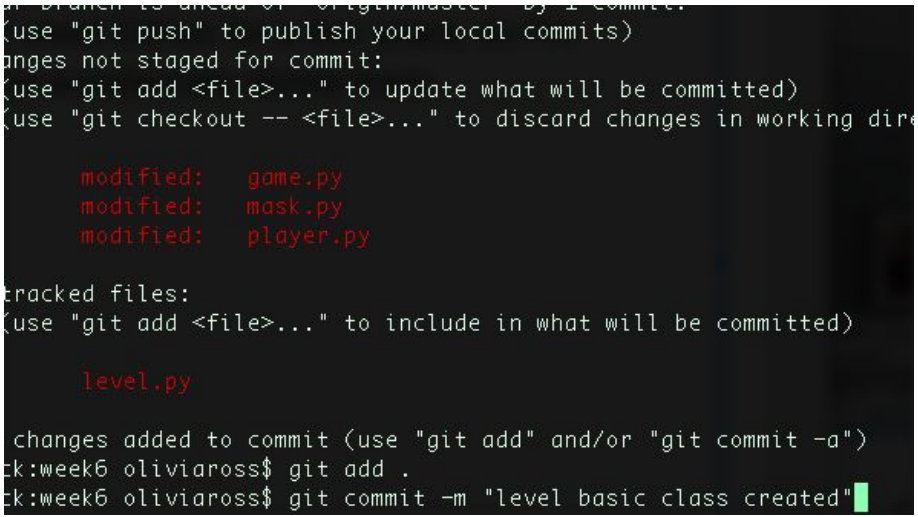
My plan to animate the players' more specific movements, like jumping, etc. involve both using sprite images as well as taking advantage of the modulo operator and data.timeCounter (a variable I will use to keep track of a personal in-game clock) to queue character animations and control their speed.

TIMELINE PLAN

MONDAY	TUESDAY	WEDNESDAY	THURSDAY
- Finish code for loading the levels - Get basic movements and render character movements - Enable player fighting (with console logging instead of animating)	- Complete MVP animations - Fill in some MVP level images and sprites - Show health point information - Prioritize pixel-based animation over tile-based	- Improve and tweak animations - Space for new things that pop up: bugs, etc. - Create start screen and directions	- Improve and tweak entire game - Fix new bugs that pop up - Add things to give polish to game

VERSION CONTROL PLAN

I’m using Github to store my code! I’ve used Github for a few years, so I’m confident I won’t lose track of my files or accidentally delete them.



STORYBOARD