Jonathan Gu

Term Project Proposal

Project Description: Name of my project: Tire of Good Luck. I'll be coding Wheel of Fortune, but I also want to implement some other key features to make it slightly different. I want to make it so the player can also rig it in favor of themselves, and have characters that they play against, just like in the game show. Additionally, the questions will be mainly based on things that have happened this year in 15112. I also will have different difficulty levels. Wheel of fortune works by players spinning a large wheel to determine how much money they win or penalties. Then, they attempt to guess a hidden word or phrase by selecting letters. Correct letter guesses are revealed in the puzzle, and you can also guess the entire word once you feel confident enough to. The player with the greatest amount of money wins the game.

Similar Projects:

Some similar projects online don't actually use animations or graphics, so I will definitely be having animations to mirror the actual game show. Additionally, some online wheel of fortune games don't actually have the feature that the letters that pop up on screen one by one and/or they don't have an option to guess letters, instead only being able to guess the whole word. I also want to implement a way for the player to rig the game for themselves, and also implement a high score system. The online wheel of fortune games don't have smart opponents, and I want to create the opponents to be extremely good at the game if needed. Additionally, having your opponents play better depending on the score total of the game.

Structural Plan:

I will be using Vscode to code my project, using multiple classes for my player and the other players that I will be playing against. I will be using a text file to store some images that I will use for my backgrounds for the project, which I learned to do at Hack112. I will have a redraw function and an onAppStart function, as well as all of the other animation functions. Additionally, I need rig game function in the player class, this way I can give the player an advantage when asked to.

Algorithmic Plan:

The trickiest part of the project will definitely be the creation of objects and traits, especially rigging it in their favor, as well as the different levels of difficulty that I want for the opponents. While a hint could be one of the easy ways to give the player an advantage, some more difficult ways will be a way to rig the wheel so that even though the animation shows that it is spinning completely equally, in reality the chances are in favor of the player. I will do this by separating the animation to the actual spin, so even though the animation looks completely fair, it isn't because I boost the chances of the good spins and decrease the chances of the bad ones. Additionally, for the different levels of difficulty, there would be the time it takes the opposing player to answer on a random interval, as well as the percentage that they would get it correct. The code would already know the answer, but the chance that the player gets it correct and the time would be different depending on what level the game is at.

Timeline Plan:

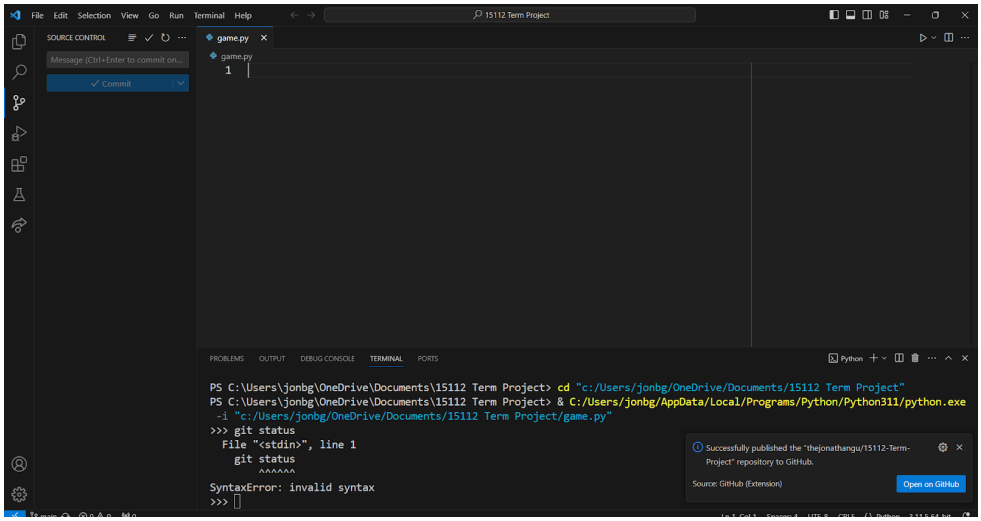
For my timeline, first I'm going to set up the project with github and VScode. This shouldn't take long at all, and during the first week I will be coding the pure wheel of fortune part of it. Difficulty, rigging, and animations will come later. I will need to make classes as well as define some of the functions that I need such as spawnLetter, chooseWord, and spinWheel.

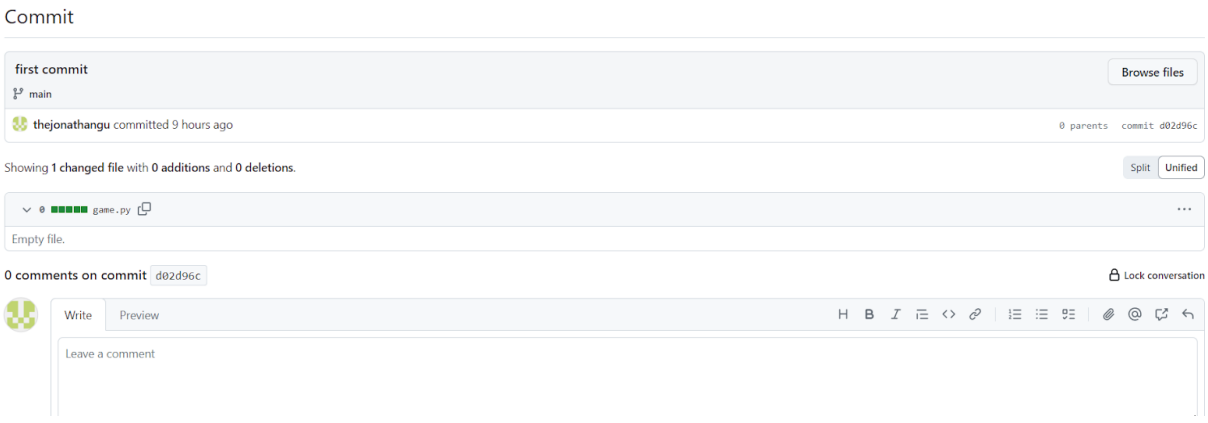
The second week, I will work on the animations part of the program. By the first week, I will have a relatively working wheel of fortune without animations. This week, I will be implementing those functions into animations. I will likely need to take inspiration especially from the game show, and reference other projects for how they would set up their animations. I will need to follow MVC and set up all the cmu graphics and drawing. This will be the MVP stage of my project.

The third week, I will work on debugging my code, as well as implementing the difficulty level and the rigging. By the end, I will submit the project (as it will be due). Debugging may take a while, as my code will likely be quite long by this point, hence why I didn't leave too much for this week to do.

Version Control Plan:

I will be using Github for version control. I already have Github, and have linked it to my VScode already. This way, I can push versions to Github whenever making changes on VScode easily.





Module List:

I'm not going to be using any outside modules

TP1 UPDATE:

I'm going to change from two computer players to one computer player. Additionally, I'm using the previous letter in each word to choose my next letter, training the ai using the 15112 syllabus for easy mode and a larger text file for hard mode using markov chains.

TP2 UPDATE:

I changed the entering letters as pressing letter boxes on the screen. I kept everything else the same as TP1.

TP3 UPDATE:

I am now using a larger text file that includes all of the 3000 most common words in the English language. Additionally, I have also used frequency tables and markov chains both to do my ai, which makes it much better at guessing what letter to guess.