Username: Magmy020

Name: Munyaradzi Magura

Student ID: 110256579

Check class:

This class manages all the checking of a guess. Which means this class will check if the guess is equal to the answer, as well as outputting the relevant information; such as square brackets if the character in the guess at a selected index mates the character at the same index within in answer, otherwise if the indexes do not match we add bars (|) around the character. If the character of guess is not in the answer then we just return the character as is.

- I added the wordDelete() method because, some variables would store the result of the previous guess or would double the characters within one guess, so this method re-sets all key variables.
- I decided to separate out the findCorrect() and findIncorrect() methods because when I was testing initially I found they would sometimes clash and cause memory leak issues. As well as create false positives in eachothers outputs i.e. answer: HELLO, result: [H][E][|L|][|L|][O]. the reason for this issue, was because the findIncorrect() method would find the first L and second L as being in the wrong place
- I added the frequency() method because if an answer had more than one character I needed a way to check how frequently that character appeared in the word. This also fixes the issue with false words such as “eeeee”

gameInstance class:

This class is responsible for running a game instance, meaning every game is one instance of this class and it calls the check class 6 times.

- the reason I separated out this class from the check class was because I felt the check class was doing too much by it self and its functions did not directly meet the scope of the game instance. Separating the two classes lessened the complexity
- .the reason I have the isLetter method, is because I needed a way for the program to figure out that the user input is a string containing only letters this is what this function does when given a guess. This finction does not check the dictionary as the words in the dictionary are considered valid by default. One problem is this function does not intuitively reject spaces in the word, however it will reject words with spaces.
- The deleteGameDetails() method, deletes all game game instance details so the key variables are free for the next game.
- I added the play() method because I needed a way to tell the werdle class to start the game.

Dictionary class:

This class contains the words and was given to us by Andrew. In my implementation this class, it is also responsible for getting a word from the list.

Werdle class:

This is the main class responsible or starting the game and getting the user to different screens such as help, statistics, and play game.

- This class has a varible named gameStats, which is a map which contains a string and an int. the string refers to how many guess the user had to make before getting the result, and the int is the number of those guess.

gameStats class:

this class is responsible for getting and setting the game statistics.

- the getGameStats method Is virtual because I wanted the werdle game class to be forced into implementing it.