

Names: Jordan Pinkava, Michael Ukenye, Luis Martinez Ferraez, Ben Jernejcic
INST326 Final Project
May 8, 2024

Documentation File:

Explanation of the purpose of each file in the repository:

1. finalProject (our main file):
 - a. This is the file that runs from the command line
2. Our text files for our themes
 - a. animals_words.txt
 - b. sports_words.txt
 - c. countries_words.txt
 - d. Fruit_words.txt
3. README.md
 - a. Ignore this file!
4. FINAL README.pdf
 - a. This is that file! Refer to this to better understand our project.
5. Exercise collab programming
 - a. For our exercise! Not relevant to the final project
6. Python file for exercise.py
 - a. For our exercise! Not relevant to the final project
7. .gitignore
 - a. ignore!

Instructions on how to run from the command line:

```
cd 326-Final-Project  
> python finalProject
```

Instructions on how to use the program and interpret the output of the program:

1. How to use the program:
 - a. Run the command line argument
 - b. Give the name of the player
 - c. Choose a theme from our theme options
 - d. Guess the word based off the list of letters given
 - i. If you guess correct, it tells you how many points you got and prompts a question about playing the game again
 1. In this case, you choose the theme and play as normal. The points are totaled in the very end once you decide to end all rounds.
 - ii. If you guess incorrectly, it'll tell you it was incorrect, how many guesses are left, and prompts you to guess again
 - e. If you choose Y to play again, you choose your theme and then play as normal

- f. If you choose N to stop playing, then it'll tell you how many points in total you got and will exit the program.
2. How to interpret the output:
 - a. Once you choose a theme, it'll give you a list of letters. Those letters are a scrambled version of the word (based on the theme), and we made a function to add extra letters, just to add challenge and complexity.
 - i. For example, an anagram of Owl would be very easy, but by adding 3-4 letters to that anagram, it adds some challenge
 - b. You refer to the list to guess the word until you either guess it correctly or run out of guesses
 - i. If you run out of guesses, you lose and the program stops completely and prompts whether or not you would like to play again
 1. If N, the program exists
 2. If Y, the program continues
 - ii. If you guess incorrectly but still have guesses, you are prompted to guess again
 - iii. If you guess correctly, it tells you how many points you received, and asks you if you'd like to play again
 - iv. If N, the program exists
 - v. If Y, the program continues

Attribution Table:

Function/Method	Primary Author	Techniques
GameState.get_past_dict	Ben	Dictionary comprehension
GameState.__str__	Ben	Magic methods
Outcome.results	Michael	Set operation
Outcome.gameplay	Gabrielle	Optional parameters
GameState.select_random_word	Luis	f-strings
GameState.select_random_word	Luis	with
parse_args	Jordan	ArgumentParser class
GameState.gameboard	Michael	max()
GameState.play_again	Jordan	Conditional Expression
Outcome.wordVerification	Gabrielle	Regex