

CSC 110 – Programming Project - Spring 2022

The Game of Wordle

Objective:

The objective of this project is to write a program in Python that will play the game of Wordle where the computer randomly chooses a five-letter word and the user has six attempts to guess the word.

The Game:

In the game of Wordle, a secret word is randomly selected by the computer. The player has six tries to guess the word. After each guess, the computer provides a hint to indicate which letters in the guess are the correct letter in the correct spot, which letters in the guess are the correct letter but in the wrong spot, and which letters in the guess are not in the word at all.

Original Wordle Game:

This game is based on the Wordle game developed by Josh Wardle that can be found here: <https://www.powerlanguage.co.uk/wordle/>. In this version of the game, a player makes a guess, and the computer colors the letters in the guess as follows:

Correct letter, correct spot = Green

Correct letter, incorrect spot = Yellow

Letter not in the word = Grey

For example, the following is a graphic representing the play of this game. The first guess was ADORE. A and E were shown (in Yellow) to be in the word, but in the wrong spots. In the second guess, A, T and E were also shown (in Yellow) to be in the word, but in the wrong spot. In the third guess, E, A and T were shown (in Green) to be in the correct spots in the word. The fourth and final guess shows that all letters are correct, in the correct spot.

A	D	O	R	E
M	A	T	E	S
W	H	E	A	T
P	L	E	A	T

Wordle Variation:

For this assignment, you will create a variation on the original Wordle game that is all text-based. Here is how it will work:

- Computer randomly chooses a word from a dictionary and asks the user to make a guess.
- User makes a guess of a 5-letter word.
- If the word is not in the dictionary, the computer asks the user to enter a different word. This does not count as a guess.
- After a valid guess, the computer displays a clue. For each letter in the guess, the computer will display one of three code letters:
 - G = Correct letter in correct spot (G for Green)
 - Y = Correct letter in incorrect spot (Y for Yellow)
 - X = Incorrect letter

For example, if the word is PLEAT and the guess is ADORE, the clue will be YXXXY.

- The computer allows the user to play again and keeps a score for all of the words played.
- The computer will not play the same word over the course of a user's game.
- The score for each game is the number of guesses it takes to get the correct word. If the user does not guess the word in 6 guesses, the score for that game is 10.

Random Words:

Your program will read in a list of 5-letter words from a dictionary:

[five-letter-words.csv](#).

Each time the user plays, the program will randomly choose a word from the list of 5-letter words. In order to allow Gradescope to test this program, we will use a random number generator with a chosen seed value. This way whenever the same seed value is used, the same words will be chosen from the list.

The program should include the following code to allow for this:

- At the top of the program:

```
import random
```

- In the definition of the main function:

```
def main(seedIn):
```

- The first line of code in the main function:

```
random.seed(seedIn)
```

- In the function that gets a random word from the list, use the `randint` function.

Computing Total Score:

The original Wordle game has one game each day. Our variation will allow the user to play as many words as they want. After each word played, the computer should ask the user if they want to play another word. It should also keep a running score for all of the words the user has played.

Testing Requirements:

You have some freedom in how you implement this program. There are two functions required to be defined as specified below so that Gradescope can test the code:

Compute the Clue:

This function will be given the user's guess and the Wordle word as parameters and it will return the clue:

```
def computeClue(guessWord, worldWord):  
    return clue
```

Main Function:

The main function should take one parameter, a seed value. This will be used to seed the random number generator so that when we test the program, we can predict which word will be chosen from the dictionary. The main function skeleton should be:

```
def main(seedValue):  
    return
```

The first line of code in the main function should be:

```
random.seed(seedValue)
```

The rest of the main function will be your code to call your other functions and run the overall program.

Sample Output

The following are examples of what your output should look like in various scenarios. Note the seed value fed to the main function in each example. This will help you test your code for the same scenario.

1. Incorrect file name

```
>>> main(1)
Please enter a file name: 5-letter-words.csv
Invalid file name try again ...
Please enter a file name: five-letter-words.csv
Make a guess: hello
HELLO
XXXXX
Make a guess: fiend
FIEND
XYXXX
Make a guess: bring
BRING
GGGXY
Make a guess: brigs
BRIGS
GGGGG

Congratulations, your wordle score for this game is 4
Your overall score is 4

Would you like to play again (Y or N)? n

Thanks for playing!
>>> |
```

2. No matching letters in first guess

```
>>> main(3)
Please enter a file name: five-letter-words.csv
Make a guess: BOOKS
BOOKS
XXXXX
Make a guess: TITLE
TITLE
XYXXG
Make a guess: SHINE
SHINE
XXGXG
Make a guess: CRIME
CRIME
GGGGG

Congratulations, your wordle score for this game is 4
Your overall score is 4

Would you like to play again (Y or N)? N

Thanks for playing!
>>> |
```

3. Matching letters but not spots in first guess

```
>>> main(2)
Please enter a file name: five-letter-words.csv
Make a guess: baker
BAKER
XYYYY
Make a guess: stake
STAKE
XXYGG
Make a guess: awake
AWAKE
GGXGG
Make a guess: awoke
AWOKE
GGGGG

Congratulations, your wordle score for this game is 4
Your overall score is 4

Would you like to play again (Y or N)? n

Thanks for playing!
>>> |
```

4. Matching letters in correct spots in first guess

```
>>> main(155)
Please enter a file name: five-letter-words.csv
Make a guess: spent
SPENT
YXXGY
Make a guess: tents
TENTS
GXYXG
Make a guess: turns
TURNS
GGGGG

Congratulations, your wordle score for this game is 3
Your overall score is 3

Would you like to play again (Y or N)? |
```

5. Wordle word with double letters

```
>>> main(4)
Please enter a file name: five-letter-words.csv
Make a guess: shade
SHADE
YXXXY
Make a guess: bests
BESTS
XYYXG
Make a guess: steps
STEPS
YXGXG
Make a guess: chess
CHESS
GXGGG
Make a guess: cress
CRESS
GGGGG

Congratulations, your wordle score for this game is 5
Your overall score is 5

Would you like to play again (Y or N)? n

Thanks for playing!
```

6. Guess with double letters

```
>>> main(2)
Please enter a file name: five-letter-words.csv
Make a guess: books
BOOKS
XXGGX
Make a guess: smoke
SMOKE
XXGGG
Make a guess: awoke
AWOKE
GGGGG

Congratulations, your wordle score for this game is 3
Your overall score is 3

Would you like to play again (Y or N)? n

Thanks for playing!
>>>
```

7. Word does not exist in dictionary

```
>>> main(792)
Please enter a file name: five-letter-words.csv
Make a guess: awoke
AWOKE
YXYXX
Make a guess: tamor
Word not in dictionary - try again...
Make a guess: colar
Word not in dictionary - try again...
Make a guess: boats
BOATS
XGYXX
Make a guess: total
TOTAL
GGGGG

Congratulations, your wordle score for this game is 3
Your overall score is 3

Would you like to play again (Y or N)? n

Thanks for playing!
>>> |
```

8. Word not guessed after six tries

```
>>> main(3)
Please enter a file name: five-letter-words.csv
Make a guess: looks
LOOKS
XXXXX
Make a guess: belly
BELLY
XYXXX
Make a guess: voice
VOICE
XXGYG
Make a guess: times
TIMES
XYYYY
Make a guess: emits
EMITS
YYGXX
Make a guess: chime
CHIME
GXGGG
Sorry, you did not guess the word: CRIME
Your overall score is 10

Would you like to play again (Y or N)? n

Thanks for playing!
>>> |
```

9. Keep a running score

```
>>> main(2)
Please enter a file name: five-letter-words.csv
Make a guess: ADOBE
ADOBE
GXGXG
Make a guess: ADORE
ADORE
GXGXG
Make a guess: AWAKE
AWAKE
GGGGG

Congratulations, your wordle score for this game is 3
Your overall score is 3

Would you like to play again (Y or N)? Y
Make a guess: THINK
THINK
YYYYX
Make a guess: FAITH
FAITH
XXYYY
Make a guess: MIGHT
MIGHT
XGGGG
Make a guess: BIGHT
BIGHT
GGGGG

Congratulations, your wordle score for this game is 4
Your overall score is 7
```

Would you like to play again (Y or N)? Y

Make a guess: ADORE

ADORE

XXXXY

Make a guess: MEETS

MEETS

XGXXX

Make a guess: HELLO

HELLO

XGGGX

Make a guess: BELLY

BELLY

GGGGG

Congratulations, your wordle score for this game is 4

Your overall score is 11

Would you like to play again (Y or N)? Y

Make a guess: CONES

CONES

XXYYX

Make a guess: NEVER

NEVER

YGXXX

Make a guess: BEANS

BEANS

XGXYX

Make a guess: FEIGN

FEIGN

GGGGG

Congratulations, your wordle score for this game is 4

Your overall score is 15

Would you like to play again (Y or N)? Y

Make a guess: PIZZA

PIZZA

XXXXY

Make a guess: ADORE

ADORE

YXXXX

Make a guess: KNOTS

KNOTS

YXXXG

Make a guess: BELTS

BELTS

XXXXG

Make a guess: SALTS

Word not in dictionary - try again...

Make a guess: BEAKS

BEAKS

XXYGG

Make a guess: PARKS

PARKS

XGXGG

Sorry, you did not guess the word: CASKS

Your overall score is 25

Would you like to play again (Y or N)? N

Thanks for playing!

>>>

Program Requirements:

Your program should meet the following requirements:

- 1) Your program should *work correctly*. Your program should do at least the following correctly:
 - a. Read in the 5-letter word dictionary from the given file.
 - b. Choose a word randomly from the dictionary.
 - c. Allow the user to make a guess.
 - d. Let the user know if the word is not in the dictionary.
 - e. Make sure that the same word is not presented twice in a game.
 - f. Provide a clue to the user when a valid word is played.
 - g. Give the user 6 tries to guess the word.
 - h. Compute a score for each word played.
 - i. Give the user an option to play again.
 - j. Compute an overall score for all words played in a game.
- 2) Your program should use good *modular design*. It should use functions for each of the main tasks of the program.
- 3) Your program should be *well-documented*. This should include your name and an overall description of the program at the top of the file. It should also include a description of any algorithms that are used in the code.
- 4) Your program should use *well-named* functions and variables. The code should be simple to read and understand what is going on given the names of the functions, and variables along with the comments in the code.

What to Submit:

Please submit your code in a file called `projWordle.py` to Gradescope in the assignment called *Programming Project - Wordle*.

NOTE: DO NOT use Gradescope to test and debug your code. You should be doing all of your testing and debugging in IDLE. Once you are confident that the code is correct, you can submit to Gradescope.