COL100 Extra Assignment 1

Due date: Saturday, 26 March

Wordle

In this assignment, you will be implementing a program that lets a user (not necessarily you) play **Wordle**.

Wordle is a popular word guessing game, where the system randomly picks a 5-letter word from a dictionary of size *n* and the user has up to six guesses in which to guess the correct **secret word**.

We will provide the list of 5-letter word list in a file (`words.txt`). Your program should load the word list and pick the secret word randomly from it. Then, it should repeatedly ask the user for guesses. After each valid guess, the program should print 2 for letters which are in the correct position, 1 for letters which are in the secret word but at the wrong position, and 0 for letters which are not in the word at all. More details are provided later in this document.

Sample program output

In this example, the program selected "`saute`" as the secret word. Text in *`italics`* is the user's input.

```
Enter attempt 1:
```
`apple`
```
10002
```

```
Enter attempt 2:
```
`acute`
```
10222
```

```
Enter attempt 3:
```

```
route
Not a valid attempt!

Enter attempt 3:
haute
02222

Enter attempt 4:
saute
YOU WIN!
```

Note that "route" was not a valid attempt as it did not contain the letter 'a', which was known to be part of the secret word.

Initialization.

The secret word has to be chosen randomly from the list of words in the text file provided to you. To do so, you can use Python's `random` module. In particular, `random.randrange(n)` will return a random integer between 0 and $n - 1$.

Program termination conditions.

Your program should repeatedly ask the user for guesses, until

1. the user correctly guesses the word, or

2. the maximum number of guesses is reached,

whichever happens earliest.

The program should print a **YOU WON!** for the correct guess. If the secret word is not found after all guesses are used up, it should print **You lose** and report what the secret word was.

For each guess, you must check that it is a valid guess according to the rules given below, otherwise reject it.

Rules for valid guesses:

The program should enforce the rules for "hard mode" in Wordle.

1. Each guess should be a word in the word list. For example, the user is not allowed to guess "`abcde`".

2. All letters found in the correct position must be used at the same position in future guesses.

3. All letters found to be part of the secret word but not at the correct position must continue to be used in future guesss.

4. The user should be allowed at most 6 guesses that satisfy the above conditions. Guesses that are not valid should not be counted.

Expected Intermediate Output.

There should be proper reporting of letter after each valid 5 letter guess. For each letter,

1. Print 2 if it is in the correct position with respect to the secret word.

2. Print 1 if it is a part of the secret word, but not at the correct position.

3. Print 0 if it is not a part of the secret word.

Efficiency.

The program should be efficient: processing each guess should not take more than $O(\log n)$ time, where n is the size of the dictionary of 5-letter words.

Your program has to represent the word list efficiently to support the operations to randomly pick a word, and to check whether a guess is a legal word in the dictionary.

Code design constraints

The program should be developed and organised in an object-oriented fashion.

- `Secret` class that encapsulates the secret word.

- `Dictionary` class that encapsulates the word list.

- Encapsulation should be used: suppose we want to change the game to 6-letter words, the changes should all be localised, and the fact that the

words are 5-letter words or the internal representation of dictionaries should not be exposed in different parts of the program.

References

- Link for the game.

- More information about the game.

- Link to Piazza Post

- Link to Moodle