Washington State University

Data Structures for Data Analytics - DA 219

Assignment 2

Professor:
your name here

Overall Assignment

In this assignment you'll work on implementing the game 'Wordle'. If you haven't played it, I suggest you try it out (a google search will help you find it!) so you know what to aim for.

Playing a game of wordle can be broken down into these steps:

1. The computer obtains a list of possible words
2. The computer selects one word as the "secret" which the player will try to guess
3. The player guesses a word
4. The computer generates a hint about which letters are correct/incorrect.
5. The hint is communicated to the player.
6. Guessing continues until the player is out of turns or wins the game.

The game is implemented in three classes: `Wordle`, `Hint`, and `WordleGame`. The `Wordle` class is responsible for items (1) and (2) above. Both the `Wordle` class and the `Hint` class are responsible for item (4). The `Wordle` class must be part of providing a hint to the player since the `Wordle` class knows what the secret word is. However, the logic for how to construct the hint itself, and the data associated with the details of the hint are encapsulated by the `Hint` class, so the `Wordle` class acts mainly as a middle-man. The `WordleGame` class contains the logic for interacting between the computer and the player. That is, it is a user of the `Wordle` and `Hint` classes. Specifically, the `WordleGame` starts steps (1) and (2) off by creating the `Wordle` object instance. The `WordleGame` allows a player to guess a word (step 3) by reading input from the keyboard, and then sends that to the computer (i.e., the `Wordle` object instance) to obtain a hint, which is then printed for the user (step 5). This "game loop" continues until no more guesses remain, or the game is won (step 6).

Getting Started

All coding takes place in the file `Wordle.py`. You should look through all the code and generally I've tried to order the parts so that easier items are implemented first. Without a doubt, the most complex code live in the `Hint` constructor. Here, you'll be constructing a Hint to the puzzle. Of course constructing a Hint requires knowing what the secret word is as well as what the guess was. Then you'll need to figure out what letters are correctly placed, what letters are in the secret word, but incorrectly placed, and what letters aren't in the secret word at all. Read the comments for the constructor for more details.

I suggest you implement this with a series of stages. First determining which characters are correctly placed then checking which are incorrectly placed, and finally determining which are not in the puzzle. Each stage will require some looping.

At the top of the Hint constructor, you should provide a comment that provides reasoning/justification for the method you've used to build the Strings each Hint instance uses.

Rubric

50% General performance of code

30% General style/approach of code

20% Reasoning and justification for Hint constructor

What to turn in:

- Hint.py
- Wordle.py
- **Note:** I will use the methods as in `WordleGame.py` to test your code