Assignment #1 – WordGame

Probably the best game never invented...

Some Useful Background Information

WordGame is a word game that tests your skill at word creation, checks if you can spell, and assesses your performance under pressure (tick... tick... tick...).

A version of WordGame for the web can be viewed (in all it's glory) and played at this web address:

<div align="center">

http://barryp.pythonanywhere.com/

</div>

When started, your WordGame program selects a word of seven or more letters from a dictionary. This word is known as the **source word**. The source word is displayed on screen, and the game records the current timestamp value.

The user of the game now has to think up seven three-or-more letter words made up from the letters contained within the source word, and they have to do this as quickly as possible. Each word is entered into the game which then – once the seven words are entered – records another timestamp value.

Upon receipt of the seven words, the game checks to ensure that:

1. each word is made up from letters (and only the letters) contained within the source word,

2. each word exists within the dictionary (i.e., it's a "real" word),

3. the words all have three letters or more,

4. there are no duplicates[1], and

5. none of the seven submitted words is the source word.

If the seven words meet the above criteria, the game computes how long the process took using the two timestamp values, recording the amount of time taken. The game then asks for the user's name to add to the *Top Scorers List*.

Upon receipt of the name, the game adds the user's name and their time into the appropriate place within the *Top Scorers List* then displays the current "Top 10" entries from the *Top Scorers List*.

And, of course, because this game is *so* cool, the user willingly accepts your kind offer to play again (or, if they're a spoilsport, they quit!).

---

1   Note: Case is not an issue here. So, "PET" is the same as "Pet" - that is, they are NOT two different words.

Points to note:

1. If one or more of the seven words are invalid, the game needs to tell the user which words were wrong, and display an appropriate "error" message.

2. If the user does not make the Top 10, they should be told where they were placed within the list (e.g., *"Nice try, Sam: You were ranked 4396th. Better luck next time."*).

Assignment Specification

1. Use Python 3 as your programming language for the game – which is to run as a web application and it to mimic the behaviour of the web-based WordGame referenced above (so you'll need Flask).

2. Be sure to test your solution (and maybe even ask your friends/family to play your game).

3. You must only submit code which has been reformatted by the BLACK code formatter.

4. Be sure to use GIT to manage your project's code, and create a new virtual environment for your project using PIPENV.

5. E-mail your solution code as a ZIP file (named per your login ID) to `paul.barry@itcarlow.ie` by the due date/time.

6. This is your first CA and is worth 15% of your final mark. While this CA is active, no new material will be presented in class (although class will still happen).

7. Due date: **Friday, November 9th 2018**. Due time: 5:00pm.

Hints & Notes

1. Be sure to read, review, and understand the `xword.py` code distributed with this assignment. This module is designed to allow you to "cheat" at crosswords, and it demonstrates some techniques that may help with your WordGame implementation. Note that the module assumes you are running Linux and that the system's default dictionary of words is installed (here's a link to the list of words my module uses: http://paulbarry.itcarlow.ie/words.txt).

2. Perhaps begin by creating a Jupyter Notebook to work out what code you need to have in place to run the mechanics of the WordGame? Once you've worked all that out, create a Flask app to deploy your WordGame as a webapp. Don't start with the webapp (as that's a bad idea).

3. Class runs as scheduled for the next 3 weeks. I'll be in attendance while you work on your assignment. You can ask me clarification questions, but if you as me a "How do I...?" type question, I'll respond with: "You'll need to write some Python code to help with that". If enough of you get stuck on something, I'll cover the "answer" on the data projector for all to see.