Computer Science 220/220L
Program 8 and Lab 10
Hangman

Learning objectives:

Develop a Python program that:

- Reads from a file
- Uses functions
- Uses decision and repetition structures to solve a problem

Part I: Assignment:

Write a program to support the children's spelling game Hangman. Call the program **`hangman.py`**. Since there are many versions of Hangman, be sure that you implement exactly the following.

The player is to guess the letters in a secret word. Use underscores to display the number of letters in the word. When the player guesses a correct letter, display the word with that letter showing. Correct guesses don't count against the player. Incorrect guesses count, and the player loses on the seventh incorrect guess. Each time the player is asked to enter a letter, the program should display how many guesses they have left and a list of all of the incorrect letters they have guessed. If the player accidentally chooses a letter that has already been guessed, this should not count as a guess.

Your program must contain at least the following functions. (If you want you may use other functions as well.)

- A function that gets a list of words from a file. Call the file **`wordlist.txt`.** You must provide your own file when you test your program.
- A function that randomly picks a secret word from the list.
- A function that plays the game (i.e., other than `main()`).
- A function that displays the blanked-out secret word.
- A function that determines whether the letters that have been entered spell the secret word (this returns a boolean).

Part II: User Interface

Once you get your functions working with text, you can add a graphical user interface of your design. The GUI should have the following elements:

- A text area where the word with underscores representing the letters of the word is presented.
- A text area where the letters that have been used are displayed. Optionally you can display the letters not chosen yet. In either case you must identify which role the displayed letters serve.
- A text area indicating the number of guesses the user has remaining.
- An area where the user enters the letter that s/he wishes to play.
- When a game is completed, a message should appear in the GUI window asking the player whether s/he wants to play another game. The GUI should provide **Yes** and **No** buttons for the player to make a choice.

Part III: The Picture

Once your basic program is working, add the hanged man. The traditional way to play Hangman is to draw a figure on a gallows. Another part of the hanged man is added for each incorrect guess. When the figure is complete, the player loses. How you draw the figure is up to you, but it should have exactly seven parts.

Submission:

This program is also an assignment for your next CSCI220L lab. You should complete the assignment and submit your finished program on or before the date specified in OAKS.