

CS 2316

Individual Homework 4 – Hangman

Due Friday September 19th, 2014

Out of 100 points

Created by Ilya Gurevich

Files to Submit: HW4.py, drawing.py

Provided Files: drawing.py, hangman.csv

For Help:

- TA Helpdesk – Schedule posted on class website.
- Email TA's or use T-Square Forums

Notes:

- **Don't forget to include the required comments and collaboration statement**
- **Do not wait until the last minute** to do this assignment in case you run into problems.

Part 1 – File I/O

In this portion of the homework you will be first gathering data from a csv file. The function name you must create is called **openFile()**. This csv file (“hangman.csv”) will contain words that will be later used in the hangman guessing game you must create. You must gather the data using any method you wish, preferably the **csv.reader()** function in the csv module (which you must import), but other methods will suffice. You must remove any extraneous characters, including the newline character (“\n”), any extra white space, and commas. Additionally you will be creating a dictionary that will hold the categories of the words you got from your csv file. Hard words are words with length 8 or higher. Medium words are words with length 4 to 7. Easy words are words with a length shorter than 4. You will then put the easy, medium, and hard words into a dictionary with the keys as the corresponding difficulties, and the values as the list of all the words.

Part 2 – Hangman

Using the dictionary that you got from the first part you will write a function named **hangman()**. The hangman program must fulfill multiple requirements

- The hangman program asks the user what kind of word they want to guess, e.g. “hard”, “easy”, or “medium”.
- The hangman chooses a word at random from the appropriate list and displays the number of letters in the word in the form of a number of underlines (spaces to be filled in) For example, if the word was “dog” then the output would be “_ _ _”
- **As long as the hangman is not complete** or the user has not guessed all the letters in the word:
 - The program asks the user to guess a letter
 - The user enters a letter and presses return
 - If the user enters more than one letter or a non-letter, tell the user to enter only one letter and try again.
 - If the user’s guess is in the word, the letter is placed in the appropriate blank; otherwise a new body part is drawn on the hangman.
 - If, after the user’s guess, the word is completely revealed, the user wins and the game is over. Report this to the user and exit.
 - If, after the user’s guess, the hangman is complete (head, body arms, legs), the user loses and the game is over. Report this to the user and exit.

In addition to the stand and some of the body parts provided to you, you must create an additional **6 more body parts**. **You can be creative!** You will add the additional drawings in the drawing.py file. In order to access the drawing methods from your HW4.py file you must import the drawing file using “**import drawing as draw**” and to access a method from the respective file you can call one using “**draw.head()**” or “**draw.leftarm()**”.

Grading:

You will earn points as follows for each function that works correctly according to the specifications.

File I/O

- Function correctly accesses hangman.csv

20

5

- Function removes any extraneous whitespace or newline characters **5**
- Function creates a dictionary with proper categories **5**
- Function create 3 different lists with appropriate word sizes **5**

Hangman **80**

- Program asks what kind of category they want **10**
- Program chooses word at random from list from specific category **10**
- Displays the number of letters in words as “_” **10**
- The program asks the user to guess a letter correctly **10**
- If the user’s guess is in the word, the letter is placed appropriately **10**
- New body parts are drawn appropriately **10**
- Programs ends on a win correctly **5**
- Program ends on a loss correctly **5**
- Correct number of extra body parts **5**
- Drawing imported correctly **5**