Carreon, Ma. Addine Anne T.
CS128-5L – A1

01/27/2024

CS128-5L - PROGRAMMING LANGUAGES FOR DATA SCIENCE LABORATORY
2Q SY2324

NAME: MA. ADDINE ANNE T. CARREON

SECTION: A1

Laboratory Exercise 6

Hangman Part 2: The Game

Now that you have built some useful functions, you can turn to implementing the function *hangman*, which takes one parameter the *secret_word* the user is to guess. Initially, you can (and should!) manually set this secret word when you run this function – this will make it easier to test your code. But in the end, you will want the computer to select this secret word at random before inviting you or some other user to play the game by running this function.

Calling the *hangman* function starts up an interactive game of Hangman between the user and the computer. In designing your code, be sure you take advantage of the three helper functions, *is_word_guessed*, *get_guessed_word*, *and get_available_letters*, that you've defined in the previous part!

Below are the game requirements broken down in different categories. Make sure your implementation fits all the requirements!

Game Requirements

A. Game Architecture:

1. The computer must select a word at random from the list of available words that was provided in words.txt. The functions for loading the word list and selecting a random word have already been provided for you in hangman.py.
2. Users start with 6 guesses.
3. At the start of the game, let the user know how many letters the computer's word contains and how many guesses s/he starts with.
4. The computer keeps track of all the letters the user has not guessed so far and before each turn shows the user the "remaining letters"

Example Game Implementation:

Loading word list from file...

55900 words loaded.

Welcome to the game Hangman!

I am thinking of a word that is 4 letters long.

Instruction

```
import random
import string

WORDLIST_FILENAME = "words.txt"


def load_words():
    """
    Returns a list of valid words. Words are strings of lowercase letters.

    Depending on the size of the word list, this function may
    take a while to finish.
    """
    print("Loading word list from file...")
    inFile = open(WORDLIST_FILENAME, 'r')
    line = inFile.readline()
    wordlist = line.split()
    print("  ", len(wordlist), "words loaded.")
    return wordlist


def choose_word(wordlist):
    """
    wordlist (list): list of words (strings)

    Returns a word from wordlist at random
    """
    return random.choice(wordlist)

# -----------------------------------

wordlist = load_words()


def is_word_guessed(secret_word, letters_guessed):
    '''
    secret_word: string, the word the user is guessing; assumes all letters are
      lowercase
    letters_guessed: list (of letters), which letters have been guessed so far;
      assumes that all letters are lowercase
    returns: boolean, True if all the letters of secret_word are in letters_guessed;
      False otherwise
    '''
    for i in secret_word:
        if i not in letters_guessed:
            return False
```

```
    return True


def get_guessed_word(secret_word, letters_guessed):
    '''
    secret_word: string, the word the user is guessing
    letters_guessed: list (of letters), which letters have been guessed so far
    returns: string, comprised of letters, underscores (_), and spaces that represents
      which letters in secret_word have been guessed so far.
    '''
    letter_guessed_string = ""
    for i in secret_word:
      if i in letters_guessed:
          letter_guessed_string+=i
      else:
          letter_guessed_string+="_"
    return letter_guessed_string


def get_available_letters(letters_guessed):
    '''
    letters_guessed: list (of letters), which letters have been guessed so far
    returns: string (of letters), comprised of letters that represents which letters have not
      yet been guessed.
    '''
    alphabet = "abcdefghijklmnopqrstuvwxyz"
    alphabet_not_guessed = ""
    for i in alphabet:
        if i in letters_guessed:
            continue
        else:
            alphabet_not_guessed+=i
    return alphabet_not_guessed


def beautify_print_word(secret_word, word):
  for i in secret_word:
      if i in word:
          print(i.lower(), end= "")
      else:
          print("_ ", end= '')
  print()


def hangman(secret_word):
    run = True
    guess = 6
    warnings = 3
    vowel_arr = ['a','e','i','o','u']
    choose_letter = ''
    guess_letter_arr = list()
    all_guess_arr = list()
    print("Welcome to the game Hangman!")
    print(f"I am thinking of a word that is {len(secret_word)} letters long.")
    print(f"You have {warnings} warnings left.")
    while(run):
        print("-"*10)
        if is_word_guessed(secret_word,guess_letter_arr):
            print("Congratulations, you won!")
            print(f"Your total score for this game is: {guess * len(guess_letter_arr)}")
            break


        if guess == 0:
            print(f"Sorry, you ran out of guesses. The word was {secret_word.lower()}")
            break


        print(f"You have {guess} guesses left.")
        print(f"Available letters: {get_available_letters(all_guess_arr)}")
        choose_letter = str(input("Please guess a letter:")).lower()


        if choose_letter not in "abcdefghijklmnopqrstuvwxyz" or len(choose_letter) != 1:
            print("Oops! That is not a valid letter.", end ='')
            if warnings <= 0:
                print(f"You have no warnings left")
                print(f"so you lose one guess:",end='')
                beautify_print_word(secret_word,guess_letter_arr)
                guess-=1
                continue
            warnings-=1
            print(f"You have {warnings} warnings left")
            continue

        if choose_letter in all_guess_arr:
            print("Oops! You've already guessed that letter.", end = '')
            if warnings <= 0:
                print(f"You have no warnings left")
                print(f"so you lose one guess:",end='')
                beautify_print_word(secret_word,guess_letter_arr)
                guess-=1
                continue
            warnings-=1
            print(f"You have {warnings} warnings left")
            beautify_print_word(secret_word,guess_letter_arr)
            continue

        if  choose_letter not in get_available_letters(get_guessed_word(secret_word,choose_letter)):
            if choose_letter not in guess_letter_arr:
                guess_letter_arr.append(choose_letter)
                all_guess_arr.append(choose_letter)
                print(f"Good guess:",end='')
                beautify_print_word(secret_word,guess_letter_arr)
                continue
        else:
            print(f"Oops! That letter is not in my word:",end='')
            beautify_print_word(secret_word,guess_letter_arr)

            all_guess_arr.append(choose_letter)
            if choose_letter in vowel_arr:
                guess-=2
            else:
                guess-=1
            continue


if __name__ == "__main__":
    temp_word = "else"
    secret_word = choose_word(wordlist)
    hangman(temp_word)
```

Code

```
Loading word list from file...
   55900 words loaded.
Welcome to the game Hangman!
I am thinking of a word that is 4 letters long.
You have 3 warnings left.
----------
You have 6 guesses left.
Available letters: abcdefghijklmnopqrstuvwxyz
Please guess a letter:a
Good guess:_ a_ _
----------
You have 6 guesses left.
Available letters: bcdefghijklmnopqrstuvwxyz
Please guess a letter:a
Oops! You've already guessed that letter.You have 2 warnings left
_ a_ _
----------
You have 6 guesses left.
Available letters: bcdefghijklmnopqrstuvwxyz
Please guess a letter:s
Oops! That letter is not in my word:_ a_ _
----------
You have 5 guesses left.
Available letters: bcdefghijklmnopqrtuvwxyz
Please guess a letter:$
Oops! That is not a valid letter.You have 1 warnings left
----------
You have 5 guesses left.
Available letters: bcdefghijklmnopqrtuvwxyz
Please guess a letter:t
Good guess:ta_ t
----------
You have 5 guesses left.
Available letters: bcdefghijklmnopqruvwxyz
Please guess a letter:e
Oops! That letter is not in my word:ta_ t
----------
You have 3 guesses left.
Available letters: bcdfghijklmnopqruvwxyz
Please guess a letter:e
Oops! You've already guessed that letter.You have 0 warnings left
ta_ t
----------
You have 3 guesses left.
Available letters: bcdfghijklmnopqruvwxyz
Please guess a letter:e
Oops! You've already guessed that letter.You have no warnings left
so you lose one guess:ta_ t
----------
You have 2 guesses left.
Available letters: bcdfghijklmnopqruvwxyz
Please guess a letter:c
Good guess:tact
----------
Congratulations, you won!
Your total score for this game is: 6
```

Winning game output

```
Loading word list from file...
   55900 words loaded.
Welcome to the game Hangman!
I am thinking of a word that is 4 letters long.
You have 3 warnings left.
----------
You have 6 guesses left.
Available letters: abcdefghijklmnopqrstuvwxyz
Please guess a letter:a
Oops! That letter is not in my word:_ _ _ _
----------
You have 4 guesses left.
Available letters: bcdefghijklmnopqrstuvwxyz
Please guess a letter:b
Oops! That letter is not in my word:_ _ _ _
----------
You have 3 guesses left.
Available letters: cdefghijklmnopqrstuvwxyz
Please guess a letter:c
Oops! That letter is not in my word:_ _ _ _
----------
You have 2 guesses left.
Available letters: defghijklmnopqrstuvwxyz
Please guess a letter:2
Oops! That is not a valid letter.You have 2 warnings left
----------
You have 2 guesses left.
Available letters: defghijklmnopqrstuvwxyz
Please guess a letter:d
Oops! That letter is not in my word:_ _ _ _
----------
You have 1 guesses left.
Available letters: efghijklmnopqrstuvwxyz
Please guess a letter:e
Good guess:e_ _ e
----------
You have 1 guesses left.
Available letters: fghijklmnopqrstuvwxyz
Please guess a letter:f
Oops! That letter is not in my word:e_ _ e
----------
Sorry, you ran out of guesses. The word was else
```

Losing game output