JLUFE Spring 2021 (Feb-July)

Final Assignment Report

JILIN UNIVERSITY OF FINANCE AND ECONOMICS

Department of College of Managment Science and Information Engineering

BSc in Information Management and Information System

(2021)

Final Assignment: Part 02

21/06/2021

MODULE: Data Mining

Submitted by: Pond & Frost(章志超 & 王凯平) 0314021805411 & 0314021805403 (1854)

QQ: 1145868752 & 577572168

RULES:

1. I have added tips and required learning resources for each question, which helps you to solve the exercise.
2. Finish the assignment in **group of two students** (**Any group find copying/sharing from other group or internet will get '0' points!!!**)
3. Once you finish the Assignment **convert your .ipynb file into PDF (https://github.com/milaan9/91_Python_Tips/blob/main/000_Convert_Jupyter_Notebook_to_PDF.ipynb** (both .pynb and .pdf file will be required!)
4. Create **.zip** file and include your two files:
   A. Your Jupyter Notebook file (**001_Python_Assignment_01.ipynb**)
   B. Your PDF converted file of 001_Python_Assignment_01.ipynb (**001_Python_Assignment_01.pdf**)
5. Name your .zip file as your student numbers and names.

   example: **0318021907632 0318021907633 Milan Nina(米兰 妮娜).zip**

Python Assignment 02

Question: Hangman Game

Write a python program to create a Hangman game.

About Game: Going back to our old school days, some of the pen-paper games were always a top for our leisure time. In Hangman user has to guess words according to the guesses determined and as soon as they lost all their wrong guesses, they were hanged (not really, but on paper 😉). In the game of Hangman, the player only has 7 incorrect guesses (head, body, 2 legs, and 2 arms, hang) before they lose the game.

Structure:

1. In Part 1, you will require to load a random word from a **dictionary (https://github.com/milaan9/92_Python_Assignments/blob/main/sowpods.txt)**.
2. In Part 2, you will require the logic for guessing the letter and displaying that information to the user.

After completing part 1 and part 2 you will need to add the following features:

Features:

- Only let the user guess 7 times, and tell the user how many guesses they have left. Example: "You have 6 guesses left!"
- No restriction in uppercase and lowercase letters.
  - Example: user can guess "a" and it will be equal to "A" or vice-versa.
- If user guesses a numbers or a special characters, don't penalize them - ask them again to choose only letter.
  - Example: user guess "9" or "?" then ask user again to choose a letter.
- If the guess letter appear more than one time in the word display it.
  - Example: Word is "Apple" and user guess the word 'p' so --> **P P** __
- Keep track of the letters the user guessed incorrectly. If the user guesses a letter they already guessed, don't penalize them - let them guess again.
- Display some picture art for the Hangman. This is challenging - do the other parts of the exercise first!
- When the player wins or loses, let them start a new game.

Expected/Similar Output:

```
**********
Welcome to Hangman!
**********
Guess one letter at a time
Game is not case sensitive


_ _ _ _ _ _ _ _

What is your guess?: a

_  A  _ _  A  _ _ _

What is your guess?: 9
Please chose just a letter: e
e is not in this word!

  ____
 |    |

You have 6 guesses left!
your previous wrong guesses: ['E']

_  A  _ _  A  _ _ _

What is your guess?: e
You have already guessed e!

_  A  _ _  A  _ _ _

What is your guess?: h
h is not in this word!

  ____
 |    |
 |    O
 |    |

You have 5 guesses left!
your previous wrong guesses: ['E', 'H']

_  A  _ _  A  _ _ _

What is your guess?: d
d is not in this word!

  ____
 |    |
 |    O
 |   /|

You have 4 guesses left!
your previous wrong guesses: ['E', 'H', 'D']

_  A  _ _  A  _ _ _
```

```
What is your guess?: b
b is not in this word!

   ____
  |    |
  |    0
  |   /|\

You have 3 guesses left!
your previous wrong guesses: ['E', 'H', 'D', 'B']

__  A  __ __  A  __ __ __

What is your guess?: k

K  A  __ __  A  __ __ __

What is your guess?: r

K  A  __ __  A  R __ __

What is your guess?: t
t is not in this word!


   ____
  |    |
  |    0
  |   /|\
  |   /

You have 2 guesses left!
your previous wrong guesses: ['E', 'H', 'D', 'B', 'T']

K  A  __ __  A  R __ __

What is your guess?: l
l is not in this word!


   ____
  |    |
  |    0
  |   /|\
  |   / \

You have 1 guesses left!
your previous wrong guesses: ['E', 'H', 'D', 'B', 'T', 'L']

K  A  __ __  A  R __ __

What is your guess?: p
p is not in this word!


   ____
```

```
   |    |
   |    |
   |    0
   |   /|\
   |   / \
__|__
```

```
You have 0 guesses left!
You lose!
your previous wrong guesses: ['E', 'H', 'D', 'B', 'T', 'L', 'P']

The word was  ['K', 'A', 'N', 'G', 'A', 'R', 'O', 'O']

Would you like to play again? [y|n]: n
```

In [1]:

```
# Solution:
import string
import random
import linecache

def isWordGuessed(secretWord, lettersGuessed):
    set1 = set(secretWord)
    set2 = set(lettersGuessed)
    if set1 <= set2:
        return True
    else:
        return False

def getGuessedWord(secretWord, lettersGuessed):
    str1 = ""
    for i in range(len(secretWord) - 1):
        str1 += "_ "
    ls = str1.split(" ")
    str1 = ""
    for it in lettersGuessed:
        if it in secretWord:
            ls[secretWord.index(it)] = it
            try:
                ls[secretWord.index(it,secretWord.index(it)+1)] = it
                ls[secretWord.index(it,secretWord.index(it)+1)] = it
                ls[secretWord.index(it,secretWord.index(it)+1)] = it
            except:
                pass

    for item in ls:
        str1 += item
        if item == "_":
            str1 += " "
    return str1

def getAvailableLetters(lettersGuessed):
    str1 = string.ascii_lowercase
    ls = list(str1)
    for i in lettersGuessed:
        if i in ls:
            ls.remove(i)
    return "".join(ls)


WORDLIST_FILENAME = "words.txt"
def loadWords():
    print("Loading word list from file...")
    a = random.randrange(1,10)
    word = linecache.getline(r'words.txt', a)
    return word

def hangman(secretWord):
    l = len(secretWord)
    print("Welcome to the game,Hangman!")
    print("I am thinking of a word that is %d letters long." %l)
    mistakesMade = 0
    lettersGuessed = ""
    while not isWordGuessed(secretWord, lettersGuessed) and (7-mistakesMade):
        print("*"*10)
```

```
    print("You have {} guesses left.".format(7-mistakesMade))
    print("Available letters: ",getAvailableLetters(lettersGuessed))
    a = input("Please guess a letter: ")
    if a in lettersGuessed:
        print("Oops! You've already guessed that letter: ",getGuessedWord(secretWord, lettersGue
    else:
        lettersGuessed += a
        if a in secretWord:
            print("Good guess:",getGuessedWord(secretWord, lettersGuessed))
        else:
            print("Oops! That letter is not in my word:",getGuessedWord(secretWord, lettersGuess
            mistakesMade += 1
    if mistakesMade == 1:
        print("________")
        print(" |    |")
    elif mistakesMade == 2:
        print("________")
        print(" |    |")
        print(" |    0")
        print(" |    |")
    elif mistakesMade == 3:
        print("________")
        print(" |    |")
        print(" |    0")
        print(" |   /|")
    elif mistakesMade == 4:
        print("________")
        print(" |    |")
        print(" |    0")
        print(" |   /|\ ")
    elif mistakesMade == 5:
        print("________")
        print(" |    |")
        print(" |    0")
        print(" |   /|\ ")
        print(" |   /")
    elif mistakesMade == 6:
        print("________")
        print(" |    |")
        print(" |    0")
        print(" |   /|\ ")
        print(" |   / \ ")
    elif mistakesMade == 7:
        print("_____")
        print(" |    |")
        print(" |    0")
        print(" |   /|\ ")
        print(" |   / \ ")
        print("__|__ ")
        print("Sorry, you ran out of guesses. The word was {}.".format(secretWord))
        #print("Do you want do this game again?")
        #result = input()
        #if result == 'y':
            #hangman(secretWord)
print("*"*10)
if isWordGuessed(secretWord, lettersGuessed):
    print("Congratulations, you won!")
    #print("Do you want do this game again?")
    #result = input()
    #if result == 'y':
        #hangman(secretWord)
```

```
    return

word = loadWords()
secretWord = word
hangman(secretWord)
#不知如何实现询问是否继续游戏
#测试成功过程中可以成功，但是成功之后无法跳出while循环
```

```
Loading word list from file...
Welcome to the game,Hangman!
I am thinking of a word that is 7 letters long.
**********
You have 7 guesses left.
Available letters:  abcdefghijklmnopqrstuvwxyz
Please guess a letter: j
Oops! That letter is not in my word: _ _ _ _ _ _
 _______
 |     |
**********
You have 6 guesses left.
Available letters:  abcdefghiklmnopqrstuvwxyz
Please guess a letter: k
Oops! That letter is not in my word: _ _ _ _ _ _
 _______
 |     |
 |     0
 |     |
**********
You have 5 guesses left.
Available letters:  abcdefghilmnopqrstuvwxyz
Please guess a letter: a
Good guess: _ a_ _ _ a
 _______
 |     |
 |     0
 |     |
**********
You have 5 guesses left.
Available letters:  bcdefghilmnopqrstuvwxyz
Please guess a letter: e
Good guess: _ a_ e_ a
 _______
 |     |
 |     0
 |     |
**********
You have 5 guesses left.
Available letters:  bcdfghilmnopqrstuvwxyz
Please guess a letter: p
Oops! That letter is not in my word: _ a_ e_ a
 _______
 |     |
 |     0
 |    /|
**********
You have 4 guesses left.
Available letters:  bcdfghilmnoqrstuvwxyz
Please guess a letter: l
Oops! That letter is not in my word: _ a_ e_ a
 _______
 |     |
```

```
 |    0
 |   /|\
**********
You have 3 guesses left.
Available letters:  bcdfghimnoqrstuvwxyz
Please guess a letter: b
Oops! That letter is not in my word: _ a_ e_ a

 _______
 |    |
 |    0
 |   /|\
 |   /
**********
You have 2 guesses left.
Available letters:  cdfghimnoqrstuvwxyz
Please guess a letter: n
Oops! That letter is not in my word: _ a_ e_ a

 _______
 |    |
 |    0
 |   /|\
 |   / \
**********
You have 1 guesses left.
Available letters:  cdfghimoqrstuvwxyz
Please guess a letter: y
Oops! That letter is not in my word: _ a_ e_ a

 _____
 |    |
 |    0
 |   /|\
 |   / \
__|__
Sorry, you ran out of guesses. The word was camera
.
**********
```

In [ ]:

```

```

In [ ]:

```

```