002_Python_Final_Assignment_02

July 12, 2021

JLUFE Spring 2021 (Feb-July)

Final Assignment Report

JILIN UNIVERSITY OF FINANCE AND ECONOMICS

Department of College of Managment Science and Information Engineering

Specialty Of Information Management and Information System

(2021)

Final Assignment: Part 02

12/07/2021

MODULE: Data Mining

Submitted by: Sehun() 0314021805421 (1854)
QQ: 2112440859 GitHub ID:sehun123456
galaxy() 0314021805419 (1854)
QQ: 1375057153 GitHub ID:galaxy1004

1 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** (both .pynb and .pdf file will be required!)
4. Create **.zip** file and include your two files:

1. Your Jupyter Notebook file (**002_Python_Assignment_02.ipynb**)
2. Your PDF converted file of 002_Python_Assignment_02.ipynb (**002_Python_Assignment_02.pdf**)

5. Name your .zip file as your student numbers and names.

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

2 Python Assignment 02

2.0.1 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**. 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!
   ____
  |    |
  |    O
  |   /|\

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!


   ____
  |    |
  |    O
  |   /|\
  |   /

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!


   ____
  |    |
  |    O
  |   /|\
  |   / \

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!


   ____
  |    |
  |    |
  |    O
  |   /|\
  |   / \
__|__
```

```
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]: LINEPERIMAGE=9
        LINES='''   ____
          |    |
```

```
        You have 6 guesses left!
           ____
          |    |
          |    O
          |    |
```

```
        You have 5 guesses left!
           ____
          |    |
          |    O
          |   /|
```

```
        You have 4 guesses left!
           ____
          |    |
          |    O
          |   /|\
```

```
        You have 3 guesses left!
           ____
          |    |
          |    O
```

```
  |   /|\
  |   /


You have 2 guesses left!
   ____
  |    |
  |    O
  |   /|\
  |   / \


You have 1 guesses left!

   ____
  |    |
  |    |
  |    O
  |   /|\
  |   / \
__|__

You have 0 guesses left!'''
```

In [2]:
```
# Solution:
import random
import string

welcome='''**********************
Welcome to Hangman!
**********************
Guess one letter at a time
Game is not case sensitive'''
print(welcome)

with open('sowpods.txt', 'r') as f:
    words = f.read().split()
num = len(words)

def print_hangman_word(mistakes=7, wrongs=[], word=''):
    lines = LINES.split('\n')
    start = mistakes * LINEPERIMAGE
    ls = []
    for line in lines[start: start + LINEPERIMAGE]:
        print(line)
    if mistakes>-1:
```

```
            print('your previous wrong guesses: %s' % wrongs)
        print(word.replace('_','__'))

    def print_word(word, guests=''):
        wk = []
        for i in range(len(word)):
            flag = True
            for g in guests:
                if g == word[i]:
                    flag = False
                    break
            if flag:
                wk.append('_ ')
            else:
                wk.append(word[i]+' ')
        wk = ''.join(wk).strip()
        if '_' in wk:
            return False, wk
        else:
            return True, wk

    while True:
        i = random.randint(0, num-1)
        word = words[i]
        guests = []
        wrongs=[]

        fail = True

        print(word)

        wk = print_word(word)[1]
        print_hangman_word(-1, wk)
        stakes = -1
        while True:
            x = input('What is your guess?: ').upper()
            guests.append(x)
            result = print_word(word, guests)
            if result[0]:
                fail = False
                break
            else:
                if wk == result[1]:
                    stakes = stakes+1
                    wrongs.append(x)
                    print_hangman_word(stakes, wrongs, wk)
                else:
                    wk = result[1]
```

```
                    print(wk.replace('_','__'))
                if stakes == 6:
                    break
            if fail:
                print('You lose!')
            else:
                print('You win!')
            x = input('Would you like to play again? [y|n]:')
            if x=='n':
                break
```

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



What is your guess?:  m


M __ __ __ __ __ __


What is your guess?:  n


M __ __ __ __ N __


What is your guess?:  b


   ____
  |    |







You have 6 guesses left!
your previous wrong guesses: ['B']
M __ __ __ __ N __


What is your guess?:  v
```

```
 ____
|    |
|    O
|    |
```

```
You have 5 guesses left!
your previous wrong guesses: ['B', 'V']
M __ __ __ __ N __

What is your guess?:  c
```

```
 ____
|    |
|    O
|   /|
```

```
You have 4 guesses left!
your previous wrong guesses: ['B', 'V', 'C']
M __ __ __ __ N __

What is your guess?:  x
```

```
 ____
|    |
|    O
|   /|\
```

```
You have 3 guesses left!
your previous wrong guesses: ['B', 'V', 'C', 'X']
M __ __ __ __ N __

What is your guess?:  z
```

```
 ____
|    |
```

```
  |    O
  |   /|\
  |   /
```

```
You have 2 guesses left!
your previous wrong guesses: ['B', 'V', 'C', 'X', 'Z']
M __ __ __ __ N __
```

```
What is your guess?:  a
```

```
   ____
  |    |
  |    O
  |   /|\
  |   /
```

```
You have 1 guesses left!

your previous wrong guesses: ['B', 'V', 'C', 'X', 'Z', 'A']
M __ __ __ __ N __
```

```
What is your guess?:  s
```

```
   ____
  |    |
  |    |
  |    O
  |   /|\
  |   / \
__|__
```

```
You have 0 guesses left!
your previous wrong guesses: ['B', 'V', 'C', 'X', 'Z', 'A', 'S']
M __ __ __ __ N __
You lose!
```

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

```
In [ ]:
```