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
                               (2020)
```

Final Assignment: Part 02 10/07/2021

MODULE: Data Mining

Submitted by: Caiden(付立斌) 0314021805413 (1854)

Arthur(杨胜杉) 0314021805402 (1854) QQ: 1224150631 GitHub ID:Arthur0520 Criss Li(李秉儒) 0314021805406 (1854) QQ: 1764077154 GitHub ID:Criss Li

1. I have added tips and required learning resources for each question, which helps you to solve the exercise.

QQ: 1318854635 GitHub ID:Caiden10086

# 3. Once you finish the Assignment convert your .ipynb file into PDF (both .pynb and .pdf file will be required!)

points!!!)

**RULES:** 

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)

2. Finish the assignment in group of two students (Any group find copying/sharing from other group or internet will get '0'

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

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

**Question: Hangman Game** 

### 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.

**Python Assignment 02** 

# Structure:

Write a python program to create a Hangman game.

 In Part 1, you will require to load a random word from a <u>dictionary</u>. 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:

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

#### • No restriction in uppercase and lowercase letters. • Example: user can guess "a" and it will be equal to "A" or vice-versa.

Features:

• 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.

• If user guesses a numbers or a special characters, don't penalize them - ask them again to choose only letter.

• Only let the user guess 7 times, and tell the user how many guesses they have left. Example: "You have 6 guesses left!"

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

- 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!
      0
You have 5 guesses left!
your previous wrong guesses: ['E', 'H']
__ A __ A __ __
What is your guess?: d
d is not in this word!
     / |
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!
 | /|\
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!
 | /|\
You have 2 guesses left!
your previous wrong guesses: ['E', 'H', 'D', 'B', 'T']
K A __ A R __ _
What is your guess?: 1
l is not in this word!
    / | \
     / \
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!
     / | \
 | / \
You have 0 guesses left!
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

# you can use more libraries if you want

In [1]: # Solution:

import random import string

people = ['''

| 111,111

| ''', '''

/| ''','''

/|\ ''','''

/ 111,111

/ \ ''', '''

/ | \

t", "ecosystem"]

def chooseword():

number = ['0','1','2','3','4','5','6','7','8','9']

0

0 / | \

```
#Function: Select a word at random from the dictionary
   i = random.randint(0, len(word)-1)
   return (word[i])
while 1:
   code = ""
    life = 7
   s=[]
   s1=""
   w = chooseword()
   word right = []
   word wrong = []
   right = []
   for i in w:
       i = i.upper()
       word_right.append(i)
       s.append("_
    s1="".join(s)
   a time\nGame is not case sensitive")
   print(s1, '\n')
    #print(word right)
    while life>0 and code!="win":
       letter = ''
       letter = input("What is your guess?:")
       while letter in number:
           letter = input("Please chose just a letter:")
       letter1=letter.upper()
       if letter1 in right:
           print("You have already guessed ",letter,'!')
       if letter1 in word_right:
           right.append(letter1)
           for i in range(len(word right)):
               if word right[i] == letter1:
                  s[i]=letter1
           if " " not in s:
               code="win"
           print("".join(s))
           continue
       if letter1 not in word_right:
           print(letter," is not in this word!")
           print(people[7-life])
           right.append(letter1)
           life = life -1
           if life==0:
               code="lose"
           print("\nYou have ", life, " guesses left!")
           word_wrong.append(letter1)
           print("your previous wrong guesses:", word_wrong)
           continue
    if code=="win":
       print("You win!")
       print("The word was ",word_right)
       choose = input("Would you like to play again? [y|n]: ")
       choose = choose.upper()
       if choose=="Y":
           continue
       else:break
    if code=="lose":
       print("You lose!")
       print("The word was ",word_right)
       choose = input("Would you like to play again? [y|n]: ")
       choose = choose.upper()
       if choose=="Y":
           continue
       else:break
******
 Welcome to Hangman!
Guess one letter at a time
Game is not case sensitive
What is your guess?:m
m is not in this word!
```

word = ["academy", "shortcut", "fitness", "famous", "revolution", "politeness", "meaning", "diploma", "enjoymen

In [ ]:

You have 6 guesses left!

You have 5 guesses left!

What is your guess?:c \_ \_\_ C\_ What is your guess?:s S\_\_ \_ C\_\_ What is your guess?:h SH\_\_ \_ C\_\_ \_ What is your guess?:o

What is your guess?:r

What is your guess?:t

What is your guess?:u

SHO\_\_ C\_\_

SHOR\_\_ C\_\_

SHORTC\_\_\_T

SHORTCUT You win!

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

0

your previous wrong guesses: ['M']

your previous wrong guesses: ['M', 'A']

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

The word was ['S', 'H', 'O', 'R', 'T', 'C', 'U', 'T']