PRT582 SOFTWARE ENGINEERING PROCESS AND TOOL

# ASSIGNMENT 1

NAME : SYED WAJEEH-UL-HASAN SHAH

STUDENT ID : 331720

GITHUB REPO LINK : <https://github.com/wajeehhasan/PRT582hangman>

## Raw Code:

import random  
source = ['invoker','project','hangman','rain','python']  
  
  
randomWord = random.choice(source)  
  
positionArray=[]  
lengthOfWord=len(randomWord)  
if lengthOfWord>1 and lengthOfWord<=4:  
 positionArray=[0,2]  
elif lengthOfWord>4 and lengthOfWord<=8:  
 positionArray=[0,2,3]  
elif lengthOfWord>8 and lengthOfWord<=12:  
 positionArray=[2,4,5,6]  
  
scrWrdList=[]  
objective\_dict={}  
hiddenWord=''  
for letter in randomWord:  
 scrWrdList.append(letter)  
  
for item in positionArray:  
 objective\_dict.update({item:scrWrdList[item]})  
 scrWrdList[item]='\*'  
for item in scrWrdList:  
 hiddenWord=hiddenWord+item  
  
life = 5  
updatedHiddenWord=''  
hiddenWord, objectives, srcWrdlist = letterRemover(randomWord)  
while(life!=0):  
  
  
 if updatedHiddenWord=='':  
 print("\nYour Challenge Word is : "+hiddenWord)  
 usrInp=input("\n\tEnter your letter : ")  
 else:  
 print("\nYour Challenge Word is : "+updatedHiddenWord)  
 usrInp = input("\n\tEnter your letter : ")  
 updatedHiddenWord=''  
 if usrInp in randomWord:  
 print("\nYour answer is right!")  
 for key,value in objectives.items():  
 if usrInp==value:  
 srcWrdlist[key]=value  
 for item in srcWrdlist:  
 updatedHiddenWord+=item  
 elif usrInp not in randomWord:  
 life = life - 1  
 print("\nYour Answer is incorrect Try later, Current Life : " + str(life))  
 if '\*' not in updatedHiddenWord:  
 if updatedHiddenWord!='':  
 print("\nCongratulations you won")  
 break

## Refactored Code

import random  
source = ['invoker','project','hangman','rain','python']  
  
#To make this game look legit we import random word from source for this we can use random liberary  
  
randomWord = random.choice(source)  
#now remove some letters so that user can guess them  
#we need to select position of letters we want to remove it will depend updon the length of the word  
#we cant just remove 1 letter from 12 letters long word it will make it obvious  
def positionSelector(rndWrd):  
 lengthOfWord=len(rndWrd)  
 if lengthOfWord>1 and lengthOfWord<=4:  
 return [0,2]  
 elif lengthOfWord>4 and lengthOfWord<=8:  
 return [0,2,3]  
 elif lengthOfWord>8 and lengthOfWord<=12:  
 return [2,4,5,6]  
#try to replace letters with \*\* sign like in passwords  
#when user try to guess them and if they guess correct it will be replaced by the letter  
def letterRemover(srcWrd):  
 positionArray=[]  
 scrWrdList=[]  
 objective\_dict={}  
 hiddenWord=''  
 for letter in srcWrd:  
 scrWrdList.append(letter)  
 #caling position function  
 positionArray=positionSelector(scrWrdList)  
 #now converting word into hidden letter form  
 for item in positionArray:  
 objective\_dict.update({item:scrWrdList[item]})  
 scrWrdList[item]='\*'  
 for item in scrWrdList:  
 hiddenWord=hiddenWord+item  
 return hiddenWord,objective\_dict,scrWrdList  
def wordInfo(): #this function will return information on the word  
 return randomWord,len(randomWord), positionSelector(randomWord)  
  
if \_\_name\_\_ == "\_\_main\_\_": #main logic and calling of above function with if and else conditions  
 life = 5  
 updatedHiddenWord=''  
 hiddenWord, objectives, srcWrdlist = letterRemover(randomWord)  
 while(life!=0):  
  
  
 if updatedHiddenWord=='':  
 print("\nYour Challenge Word is : "+hiddenWord)  
 usrInp=input("\n\tEnter your letter : ")  
 else:  
 print("\nYour Challenge Word is : "+updatedHiddenWord)  
 usrInp = input("\n\tEnter your letter : ")  
 updatedHiddenWord=''  
 if usrInp in randomWord:  
 print("\nYour answer is right!")  
 for key,value in objectives.items():  
 if usrInp==value:  
 srcWrdlist[key]=value  
 for item in srcWrdlist:  
 updatedHiddenWord+=item  
 elif usrInp not in randomWord:  
 life = life - 1  
 print("\nYour Answer is incorrect Try later, Current Life : " + str(life))  
 if '\*' not in updatedHiddenWord:  
 if updatedHiddenWord!='':  
 print("\nCongratulations you won")  
 break

## Test Cases :

import unittest  
from hangman\_proj import source, randomWord,positionSelector,wordInfo  
  
class TestStringMethods(unittest.TestCase):  
  
 def randomWordSelected(self):  
 self.assertIn(randomWord,source,'Word is not in the list')  
  
 def letterPositioningCheck(self):  
 self.assertEqual(len(positionSelector('wajeeh')), 3, "positioning failed either removing less or more letters")  
 def returnThree(self):  
 self.assertTrue(len(wordInfo())==3,'All information is not returned')  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 unittest.main()

# Screenshot of Working Code:

# Case 1 Pass : C:\Users\The Chosen One\AppData\Local\Microsoft\Windows\INetCache\Content.Word\hangmanpass.png

## Case 2 Fail : C:\Users\The Chosen One\AppData\Local\Microsoft\Windows\INetCache\Content.Word\hangmanfail.png

## Test Case :



The above code is refactored version,

* Before this no functions were used and every line of code was inter-dependent on each other.
* Making changes were very difficult as it will affect the code as a whole.
* So, we needed to make block of code so it become more reusable and general in nature
* Code divided into,
  1. PositionSelector
     + this block defines how many letter will be replaced from each given word and return the coordinate of removable letters in the shape of an array
  2. letterRemover
     + This will take random word out of list of word defined as source in program and convert it into like for example “rain” into “r\*i\*n”
  3. wordInfo
     + this return all the information of the word that user is guessing so it can be used in main logic

Simplicity of the code can be seen in the snap below,

# Simplicity of code :

