

Assignment5: Hangman

- explanation

Date: 12/03/2023

Name: Mengyuan Li

Group_16 members:

- Gimhani Kaushalya Garusing Arachchige
- Xin Yuan
- Mengyuan Li
- Le Phuoc Hoang Dan
- Xiuru Kang

Code Repository:

- <https://github.com/Mengyuan112/Assignment5> (<https://github.com/Mengyuan112/Assignment5>)

First, the game requires you to enter your name; Enter a letter after starting the game. If the letter is one of the randomly given words, the letter will be told where the letter is in the word; If the letter you give is not in this word, the game will tell you that you guessed wrong, and you have several chances to guess, and also draw the pattern of hanging people upside down; If you guessed every letter correctly, you would win; If you guess wrong every time, you will lose the game, and you will be hanged upside down

In []:

```
import random
import time
```

- Import random module to generate random index to randomly pick a word from word list
- Import time to put program to load for certain amount of time instead of instantly loading everything. It makes program more robust and looks like a real game

In []:

```
# Initial Steps to invite in the game:
print("\n Hello! Welcome to Hangman game\n")
name = input("Enter your name: ")
```

- Get input name and assign it to variable called name

In []:

```
print("Nice to meet you " + name )
time.sleep(2)
print("Game is starting...\n ")
time.sleep(3)
```

- Make program to pause for 2 (3) seconds then continue to the next command

In []:

```
def main():
    global count
    global display
    global word
    global already_guessed
    global length
    global play_game
    words_to_guess = ["mother", "passion", "eternity", "fantastic", "destiny", "liberty", "peace", "suns",
                     , "blue"]
```

- Given some guessing words

In []:

```
word = random.choice(words_to_guess)
length = len(word)
count = 0
display = ' _ ' * length
already_guessed = []
play_game = ""
```

- Give words randomly

In []:

```
def play_loop():
```

- this function is to check if the player wants to play again or not

In []:

```
global play_game
```

- we check the player's answer and set it as value and store it inside play game variable

In []:

```
play_game = input("Do You want to play again? y = yes, n = no \n")
```

- This is to take input from user

In []:

```
while play_game not in ["y", "n", "Y", "N"]:
    play_game = input("Do You want to play again? y = yes, n = no \n")
if play_game == "y":
    main()
elif play_game == "n":
    print("Thanks For Playing! ")
    exit()
```

- this while loop is to check what the player wants. If he/she chooses y or Y, we run main() function. If he/she chooses n or N, we print out the message and then stop the program with the exit command.

In []:

```
def hangman():
```

- This hangman() function is the main game play

In []:

```
global count
global display
global word
global already_guessed
global play_game
```

- These are global variables. They are set to global so that they can be accessed from other functions or classes.

In []:

```
limit = 5
```

- This is to limit the number of times that the player can play. And we set it to 5 times maximum.

In []:

```
guess = input("This is the Hangman Word: " + display + " Enter your guess: \n")
```

- this is where the player guesses words, and we store it inside the guess variable.

In []:

```
guess = guess.strip()
```

- This strip method is to remove the whitespace from the beginning and the end of the string

In []:

```
if len(guess.strip()) == 0 or len(guess.strip()) >= 2 or guess <= "9":
    print("Invalid Input, Try a letter\n")
    hangman()
```

- This condition is to check whether the player's input is valid. If no input, meaning length of input is 0: invalid. If length of input after removing whitespaces is more than 2: invalid. If length of input after removing whitespaces is less than or equal to 9: invalid. After, we print the message. Then, run the hangman() function again.

In []:

```
elif guess in word:
```

- This is to loop through word, and check if guess (the player's guess) is in that word.

In []:

```
already_guessed.extend([guess])
```

- The extend() method is to add the guess into an array.

In []:

```
index = word.find(guess)
```

- the find method is to check if the guess is in that word.

In []:

```
word = word[:index] + "_" + word[index + 1:]
```

- we have an array, and set its variable name is word. Then we add each letter to that word array. word[:index] : mean from index 0 to and less than index.

In []:

```
display = display[:index] + guess + display[index + 1:]
print(display + "\n")
```

- we will display what the player have guessed


```
print("      \n"  
      |      | \n"  
      |      | \n")
```

```
    " |      | \n"
    " |      0 \n"
    " |     /\ \ \n"
    " |     /\ \ \n"
    "_|_ \n")
print("Wrong guess. You are hanged!!!\n")
print("The word was:", already_guessed, word)
play_loop()
```

```
if word == '_' * length:
    print("Congrats! You have guessed the word correctly!")
    play_loop()

elif count != limit:
    hangman()
```

main()

hangman()