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

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 robustand 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 wantsIf he/she choose y or Y, we run main() functionIf he/she shoose n or N, we print out the messageand 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 funcstion or classes

In [ ]:

```
    limit = 5
```

- This is to limit number of times that 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 player quess words, and we store it inside 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'sinput is validIf no input, meaning length of input is 0: invalidor length of input after removing whitespaces ismore than 2: invalidor length of input after removing whitespaces isless than or equal to 9: invalidAfter, we print the messageThen, run the hangman() function again

In [ ]:

```
elif guess in word:
```

- This is to loop through word, and check if guess (the player’s guess in 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:]
```

- display = display[:index] + guess + display[index + 1:]