

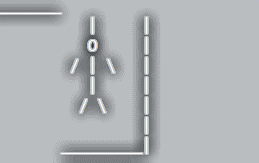
School of Mechanical & Manufacturing Engineering SMME

Department of Aerospace Engineering NUST

Python Project

Hangman

Designed by Ayesha Ayaz

****

**Course** Fundamental of Programming II

**Course code**: CS 116

# Hangman

# Objective

The objective of this project was to develop the hangman game in Python. The game as we all know requires the player to guess the random word from list of many other random words within a limited number of attempts (8 here in this case). It involes interactive gameplay with visual feedback using a series of hangman stages further developing my python skills and understanding about syntax codes and especially importance of indentations

# 

# Introduction

The hangman game is a classic word guessing game where the player attempts to decode all the letters of the word. For each incorrect guess a part of the hangman figure is drawn. The game finally ends if the whole man is drawn or the player has guessed the word correctly

The program implements hangman using python, understanding core programming concepts like loops sets conditional statements and input handling. It was important for me to breakdown hangman into small tasks including

* Selecting a random word.
* Validating user inputs.
* Tracking guessed and remaining letters.
* Updating the game state.

And then further diving into control flow how each task will be completed and the path it will follow

Further along I will be explaining the code in more depth and break it down for easier understanding additionally we will also discuss the error codes I faced going along making this project. It was harder to implement the logic then you can imagine but try and error was my go-to, though thankfully when I completed the code only syntax error were most of my concerns.

**CODE BREAKDOWN AND ANALYSIS**

Defining function to visual hangman at each stage

Defining a function to pick a random word for all the words

List of all the random words

Importing random module

import random

allwords = ["aerospace", "ayesha", "spain", "tomorrow x together"]

def chooseword():

return random.choice(allwords)

def hangmandisplay(tries):

stages = [

"""\_\_\_\_\_\_\_\_

| |

O |

/ | \\ |

| |

/ \\ |

|

\_\_\_\_\_\_\_\_\_|""",

"""\_\_\_\_\_\_\_\_

| |

O |

/ | \\ |

| |

/ |

|

\_\_\_\_\_\_\_\_\_|""",

"""\_\_\_\_\_\_\_\_

| |

O |

/ | \\ |

| |

|

|

\_\_\_\_\_\_\_\_\_|""",

"""\_\_\_\_\_\_\_\_

| |

O |

/ | \\ |

|

|

|

\_\_\_\_\_\_\_\_\_|""",

Main game logic

*word* is the word to be guessed

*wordletters* are all the letters of the word

*guessedletters* are all the letters the player has guessed

*tries* are the number of tries the player has

Printing what the player will be welcomed to including the hangman display at 0 tries and the number of letters the word is signified by “\_”

"""\_\_\_\_\_\_\_\_

| |

O |

/ | |

|

|

|

\_\_\_\_\_\_\_\_\_|""",

"""\_\_\_\_\_\_\_\_

| |

O |

| |

|

|

|

\_\_\_\_\_\_\_\_\_|""",

"""\_\_\_\_\_\_\_\_

| |

O |

|

|

|

|

\_\_\_\_\_\_\_\_\_|""",

"""\_\_\_\_\_\_\_\_

| |

|

|

|

|

|

\_\_\_\_\_\_\_\_\_|"""

]

return stages[tries]

def hangman():

word = chooseword().upper()

wordletters = set(word)

guessedletters = set()

tries = 7

print("Welcome to Hangman!")

print(hangmandisplay(tries))

print("\_ " \* len(word))

To show progress

Used the .join command to join the set into a string rather than a set

Displays the unguessed words as underscores and guessed words as letters

Game loop that runs as long as the no. of tries is greater than 0 and there are still letters to guess

while tries > 0 and wordletters:

currentword = [letter if letter in guessedletters else "\_" for letter in word]

print("\nCurrent word:", " ".join(currentword))

guess = input("Guess a letter: ").upper()

if len(guess) != 1 or not guess.isalpha():

print("Please enter a single alphabet.")

continue

if guess in guessedletters:

print(f"You've already guessed '{guess}'. Try again.")

elif guess in wordletters:

print(f"Good job! '{guess}' is in the word.")

guessedletters.add(guess)

wordletters.remove(guess)

else:

print(f"'{guess}' is not in the word. You lose a try.")

guessedletters.add(guess)

tries -= 1

print(hangmandisplay(tries))

if not wordletters:

print(f"\nCongratulations! You've guessed the word: {word}")

else:

print(f"\nGame over! The word was: {word}")

hangman()

Takes user input converts into upper letter: further validating the input as a single alpha unit

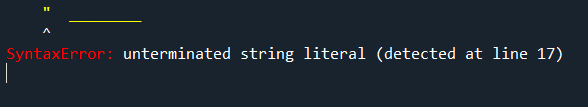
Calls Hangman function to start the game

Displays hangman per turn and finally the win-lose outcome . win if player guesses the word or lose if tries reaches 0

Checks the guesses if its already guessed or if it’s a correct guess or a wrong guess which decreases tries

Uses f-string to create string interpolation for cleaner and efficient look

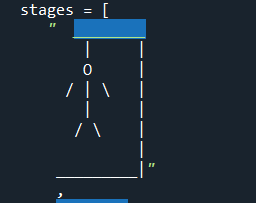
# ERRORS



Error occurred in (look at figure 1.0 and 2.0)

To fix this I realized I need to triple quote it so all objects appear as string as a multi line string

*Figure 1.0*

A backlash basically signifies an escape sequence which was leading to error, python expects something like \\ to represent a literal \ (look at figure 2.0 and 3.0)

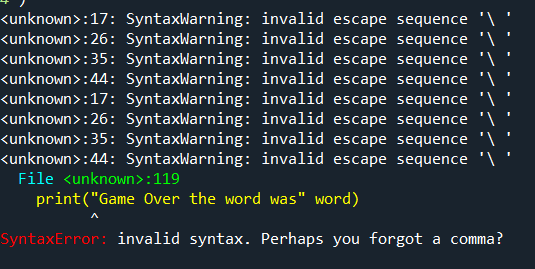
Other errors included

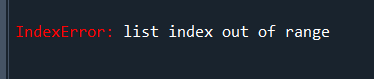
A lot of indentation errors which I corrected by proper indentation

(figure 4.0)

Another error I faced was when I was trying to print currentword it was printing as a list to fix this I had to display it as a string so I used the .join command

*Figure 2.0*

Along that I also had trouble with the logic when player guessed an incorrect alphabet my code kept taking the incorrect input I was able to solve it when I finally remembered we use continue at the end of the loop to skip the rest of the loop



*Figure 3.0*

*Figure 4.0*