

HANGMAN GAME REPORT

INT 213:PYTHON PROJECT



NAME:K.Shanmukh Sai Vinayak

REG NO:12112432

ROLL NO:RK21YKA054

SUBMITTED TO:NAVPREET RUPAL

ABSTRACT

- This is a simple Hangman game using Python programming language. We can use this as a small project to boost their programming skills and understanding logic. The Hangman program randomly selects a secret word from a list of secret words. The random module will provide this ability, so line 1 in program imports it. Hangman is a popular word game in which one player (the "chooser") chooses a secret word and another player (the "guesser") attempts to guess the word one letter at a time. If a guessed letter appears in the word, all instances of it are revealed. If not, the guesser loses a chance. If the guesser figures out the secret word before he or she runs out of chances, he or she wins. If not, the player who chose the word wins.

About Project

- This is a simple Hangman game using Python programming language. Beginners can use this as a small project to boost their programming skills and understanding logic.
- 1. The Hangman program randomly selects a secret word from a list of secret words. The random module will provide this ability, so line 1 in program imports it.
- 2. The Game: Here, a random word (a fruit name) is picked up from our collection and the player gets limited chances to win the game.
- 3. When a letter in that word is guessed correctly, that letter position in the word is made visible. In this way, all letters of the word are to be guessed before all the chances are over.
- 4. For convenience, we have given 10 chances. For example, word to be guessed is blue so we can give 10 times wrong letter. After that you will lose the game.

CODE

```
import random

from tkinter import *

from tkinter import messagebox


score = 0

run = True


# main loop
while run:

    root = Tk()

    root.geometry('905x700')

    root.title('HANG MAN')

    root.config(bg = '#E7FFFF')

    count = 0

    win_count = 0


# choosing word
index = random.randint(0,4)
```

```
file = open('rohan.txt','r')

l = file.readlines()

selected_word = l[index].strip('\n')


# creation of word dashes variables

x = 250

for i in range(0,len(selected_word)):

    x += 60

    exec('d{ }=Label(root,text="_",bg="#E7FFFF",font=("arial",40)).format(i))

    exec('d{ }.place(x={ },y={ }).format(i,x,450))


#letters icon

al = ['a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z']

for let in al:

    exec('{ }=PhotoImage(file="{ }.png").format(let,let))


# hangman images

h123 = ['h1','h2','h3','h4','h5','h6','h7']

for hangman in h123:

    exec('{ }=PhotoImage(file="{ }.png").format(hangman,hangman))
```

#letters placement

```
button =  
[[['b1','a',0,595],[b2','b',70,595],[b3','c',140,595],[b4','d',210,595],[b5','e',280,595],[b6','f',350,595],[b7','g',420,595],[b8','h',490,595],[b9','i',560,595],[b10','j',630,595],[  
'b11','k',700,595],[b12','l',770,595],[b13','m',840,595],[b14','n',0,645],[b15','o',70,645],[b16','p',140,645],[b17','q',210,645],[b18','r',280,645],[b19','s',350,645],[b20','t',  
420,645],[b21','u',490,645],[b22','v',560,645],[b23','w',630,645],[b24','x',700,645],[b25','y',770,645],[b26','z',840,645]]
```

for q1 in button:

```
exec('{}=Button(root,bd=0,command=lambda:check("{}","{}"),bg="#E7FFFF",activebackground="#E7FFFF",font=10,image={}).format(q1[0],q1[1],q1[0],q1[1]))  
    exec('{} .place(x={},y={}).format(q1[0],q1[2],q1[3]))
```

#hangman placement

```
han = [['c1','h1'],[c2','h2'],[c3','h3'],[c4','h4'],[c5','h5'],[c6','h6'],[c7','h7']]
```

for p1 in han:

```
    exec('{}=Label(root,bg="#E7FFFF",image={}).format(p1[0],p1[1]))
```

placement of first hangman image

```
c1.place(x = 300,y =- 50)
```

```

# exit buton
def close():
    global run
    answer = messagebox.askyesno('ALERT','YOU WANT TO EXIT THE GAME?')
    if answer == True:
        run = False
        root.destroy()
e1 = PhotoImage(file = 'exit.png')
ex = Button(root,bd = 0,command = close,bg="#E7FFFF",activebackground = "#E7FFFF",font = 10,image = e1)
ex.place(x=770,y=10)
s2 = 'SCORE:'+str(score)
s1 = Label(root,text = s2,bg = "#E7FFFF",font = ("arial",25))
s1.place(x = 10,y = 10)

# button press check function
def check(letter,button):
    global count,win_count,run,score
    exec('{ }.destroy()'.format(button))
    if letter in selected_word:
        for i in range(0,len(selected_word)):
            if selected_word[i] == letter:
                win_count += 1
                exec('d{ }.config(text="{ }")'.format(i,letter.upper()))

```

```
if win_count == len(selected_word):
    score += 1
    answer = messagebox.askyesno('GAME OVER','YOU WON!\nWANT TO PLAY AGAIN?')
    if answer == True:
        run = True
        root.destroy()
    else:
        run = False
        root.destroy()
else:
    count += 1
    exec('c{ }.destroy()'.format(count))
    exec('c{ }.place(x={ },y={ })'.format(count+1,300,-50))
    if count == 6:
        answer = messagebox.askyesno('GAME OVER','YOU LOST!\nWANT TO PLAY AGAIN?')
        if answer == True:
            run = True
            score = 0
            root.destroy()
        else:
            run = False
            root.destroy()
root.mainloop()
```

RESULT:

HANG MAN

SCORE:0

EXIT

— — — — —

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

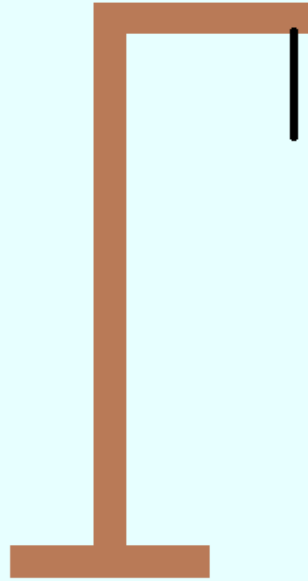
Y

Z

This is the starting of the hangman game. Here we can see there are six blanks, so it means we have to guess a six words letter by clicking on the alphabet buttons.

SCORE:0

× EXIT



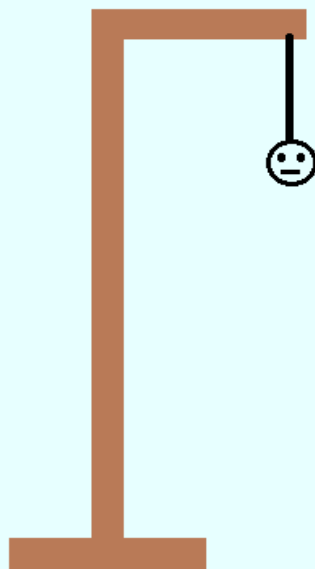
Y _ _ _ _



If we click the right alphabet that is in the word which we are guessing then it will fill the appropriate blank.

SCORE:0

× EXIT



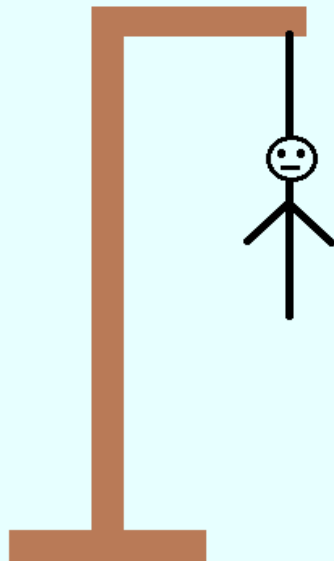
Y E _ _ _ _



See if we enter a wrong alphabet the Hangman starts to build

SCORE:0

× EXIT



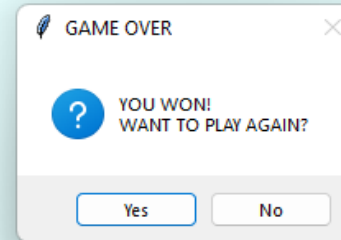
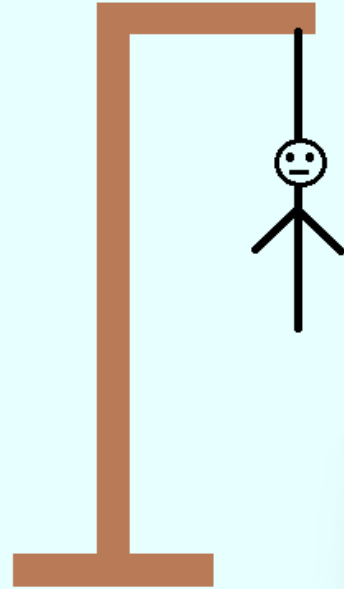
Y E L L _ _



It will build in steps whenever you click wrong alphabet.

SCORE:0

× EXIT



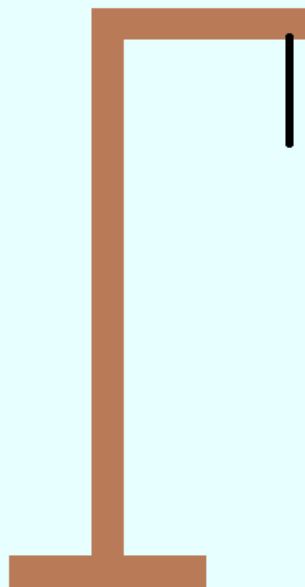
Y E L L O W



After guessing the word correctly, it will show that you won the game. Moreover it will ask you want to play again or exit.

SCORE:1

× EXIT



_ _ _



If you click yes then the game will start again and the score will increased by 1.

CONCLUSION

- In the conclusion of this project, Hangman is a traditional game, typically played with words. It's possible, however, to play Category Hangman rather than guessing words the player might guess names of cities, or athletes, or fictional characters, or Duke professors, or top forty song titles the list is endless. You'll be writing a program to play a "guess a word letter-by-letter" version of hangman as shown above. You'll also be doing some statistical analysis of the words used in the Hangman game.

REFERENCES

<https://www.youtube.com/>

<https://www.google.com/>

<https://www.wikipedia.com/>

THANKYOU