**GIET INSTITUTION**

**A Project Report**

**on**

**Hang Your Brain**

**Submitted For The Partial Fulfillment Of**

**Certificate Course On Python Programming**

**by**

**Sri Mani Neha Karamcheti,20551A4624**

**Yagneswar Sanipini,20551A4649**



### **LEARNING AND DEVELOPMENT DEPARTMENT**

**GODAVARI INSTITUTE OF ENGINEERING & TECHNOLOGY(A)**

**Chaitanya Knowledge City,** **NH-16, Rajahmundry 533296, AP**

**Title of the Project:** Hang Your Brain

**Abstract:**

Hangman is a popular word guessing game where the player attempts to build a missing word by guessing one letter at a time. After a certain number of incorrect guesses, the game ends and the player lose the game. The game also ends if the player correctly identifies all the letters of the missing word.

**Introduction:**

1. **About Python:**

Python is a popular general-purpose programming language. It is used in machine learning, web development, desktop applications, and many other fields. Fortunately for beginners, Python has a simple, easy-to-use syntax. This makes Python a great language to learn for beginners.

1. **About Tkinter:**

**Tkinter:**

Tkinter is the Python interface to the Tk GUI toolkit shipped with Python.

**geometry ():**

This method is used to set the dimensions of the Tkinter window and is used to set the position of the main window on the user’s desktop.

**Button:**

The Button widget is used to add buttons in a Python application. These buttons can display text or images that convey the purpose of the buttons.

You can attach a function or a method to a button which is called automatically when you click the button.

**Syntax:**

***w = Button (master, option=value, ...)***

**Messagebox:**

MessageBox Widget is used to display the message boxes in the python applications. This module is used to display a message using provides a number of functions.

**Syntax:**

***messagebox.Function\_Name(title, message [, options])***

**destroy ():**

destroy is a universal widget method i.e we can use this method with any of the available widgets as well as with the main tkinter window.

**Syntax:**

***widget\_object = Widget (parent, command = widget\_class\_object.destroy)***

**Game Description:**

1. **Pen and Paper Hangman:**

**Number of Players:** 2 or more

**Materials:**

Hangman Game Board, white board marker and a book just read paper and pencils may be used.

**Object:**

One player thinks of a word and the other tries to guess it by guessing letters. Each incorrect guess brings you closer to being "hanged." This game helps to sharpen children’s spelling and word-decoding skills.

**Directions**:

Decide who is going first. Start the game by having this person choose a word or phrase in their mind. (Choose a word from the book just read, and for quick reference to the word, use a bookmark to keep track of the page it is located on.)

Place one dash on the bottom of the game board for each letter of the word or words chosen. Leave a space between words.

If the word is dog, draw three spaces, like this: \_\_ \_\_ \_\_.

Have the other player guess one letter at a time - or he or she can use a turn to guess the entire word or words.

Fill in the letter everywhere it appears on the appropriate dash (or dashes) each time the person guesses correctly. Circle the letter on the alphabet if is guessed correctly. Add one body part to the drawing each time the letter chosen is not in the word. Begin by drawing a head attached to the short vertical line (the "noose"). Add eyes, ears, nose, hair, body, legs, and arms. Put an X through the letter that was guessed and not correct. You may also wish to make your drawings very elaborate - one ear at a time, a neck, and a belly button - so that children will have a lot of guesses before losing.

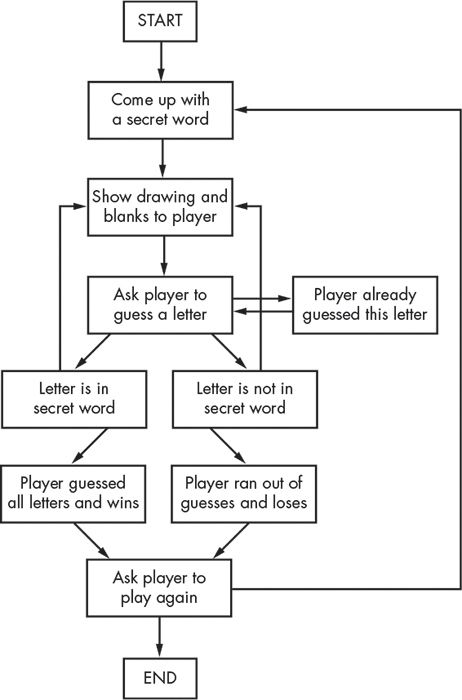
If the drawing of the person is completed before the word or words are guessed, the guessing player loses. If the player figures out the word or words first, he or she wins.

1. **Online Hangman:**

**How to play hangman:**

1. Hangman is a simple word guessing game. Players try to figure out an unknown word by guessing letters. If too many letters which do not appear in the word are guessed, the player is hanged (and loses).
2. Setup the game by drawing a gallow and an underline for each letter in the unknown word.
3. As letters in the word are guessed, write them above the corresponding underline.
4. If a letter not in the word is guess, draw a picture of a person on the gallow–one part for each incorrect letter guess.
5. Most frequently, the person is drawn in 6 parts (for 6 letter guesses) in the order: head, body, left leg, right leg, left arm, right arm.

**Flowchart:**



**Testing:**

**Source 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('HANGMAN')

root.config(bg = 'white')

count = 0

win\_count = 0

# Choosing word

index = random.randint(0,1000)

file = open('words.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="white",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('{}=Photo Image(file="{}.png")'.format(let,let))

# Hangman images

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

for hangman in h123:

exec('{}=Photo Image(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="white”, activebackground="white”, font=10, image= {})’. format(q1[0], q1[1], q1[0], q1[1]))

exec ('{}. place (x= {}, y= {})’. format(q1[0], q1[2], q1[3]))

#Placing Hangman

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

for p1 in han:

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

# Placing first hangman image

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

# Exit buton

def close ():

global run

answer = messagebox.askyesno('ALERT','DO 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="white",activebackground = "white",font = 10,image = e1)

ex.place(x=770,y=10)

s2 = 'Score:'+str(score)

s1 = Label (root,text = s2,bg = "white",fg="black",font = ("arial",25))

s1. place (x = 10, y = 10)

# Check function for button

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 +=10

answer = messagebox.askyesno('GAME OVER','YOU WON!\n DO YOU WANT 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()

la = Label (root, text = "Developed by Neha and Yagneswar”, fg="green")

la.config(font =("Courier", 14))

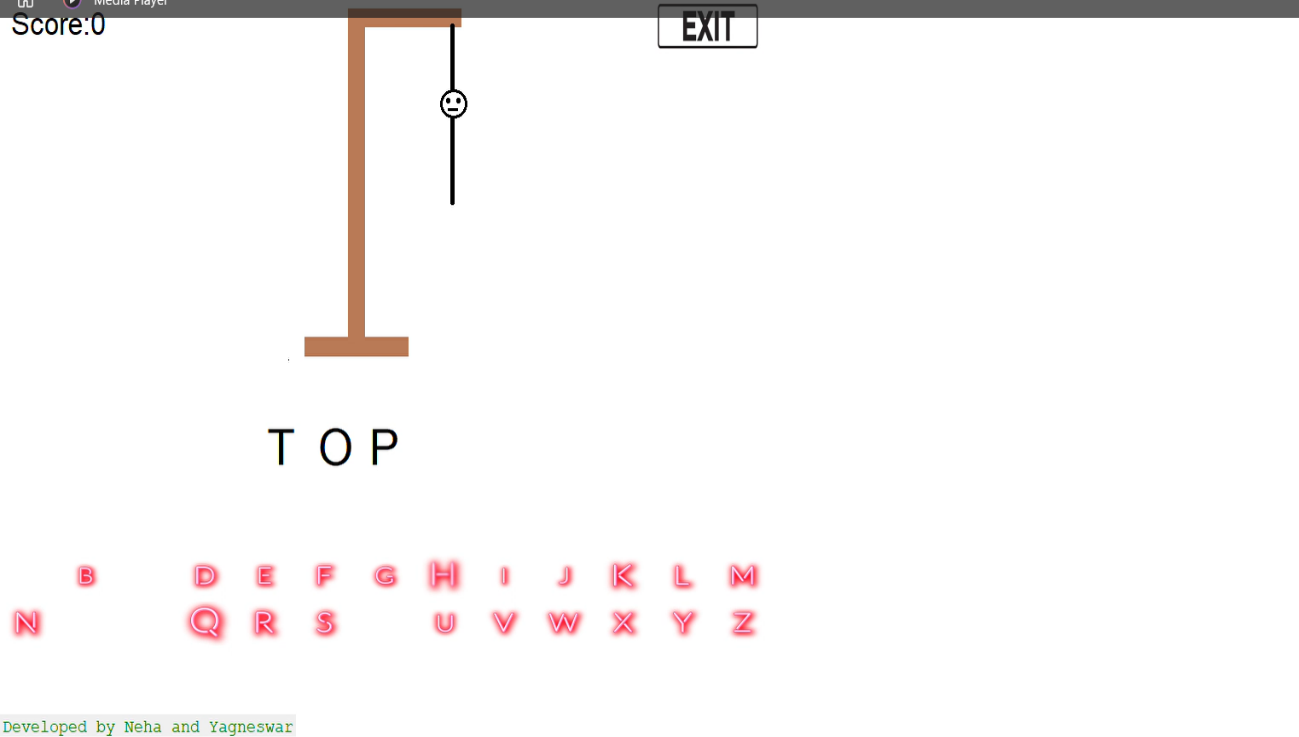
la.grid(row=1, column=0, sticky=S)

root.grid\_rowconfigure(1,weight=1)

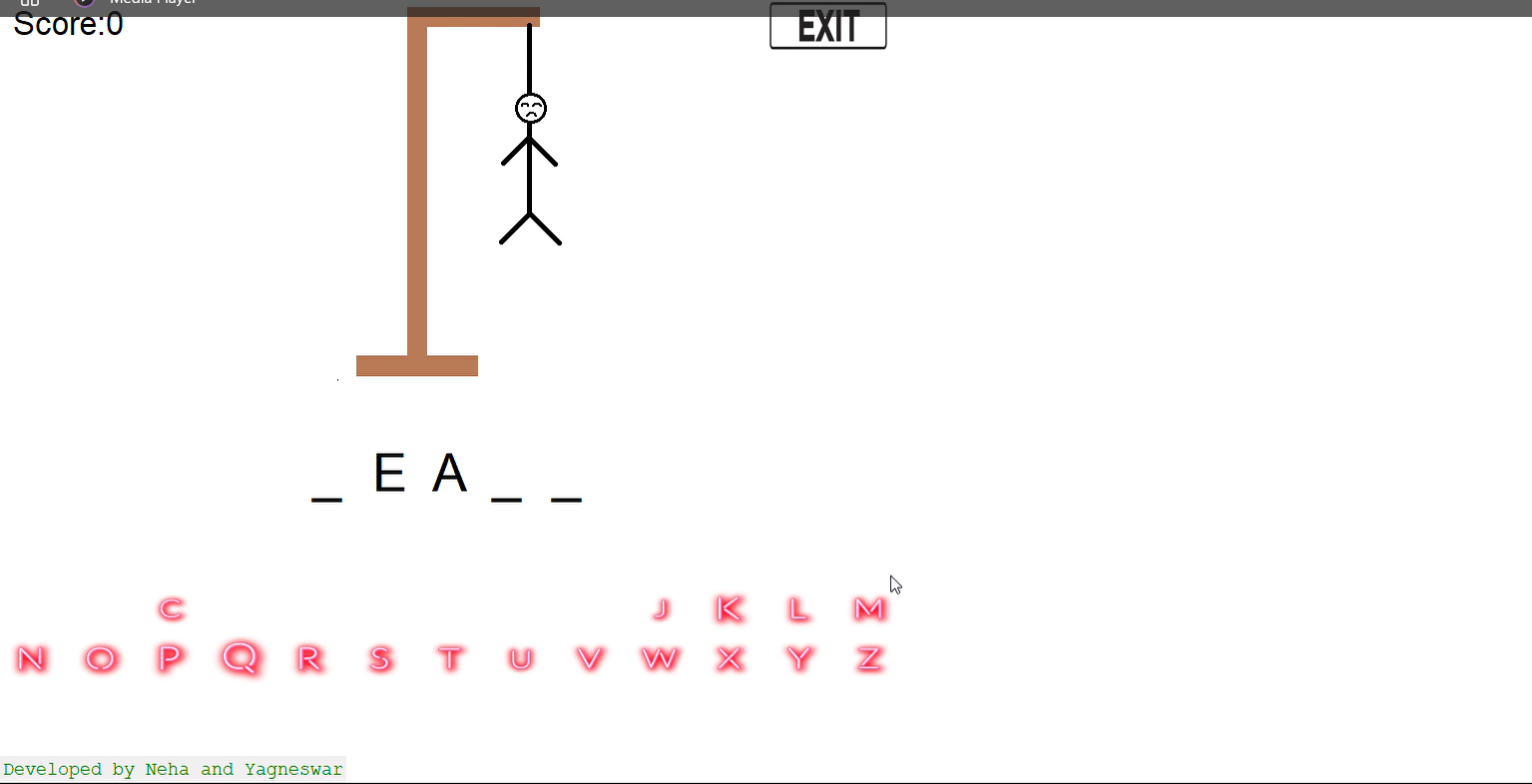
root.mainloop()

**Output:**

*Winning of game and score increases by 10*

****

*Losing of game*



**Conclusion:**

There are many games like pubg, cod and many more that games are actually good at time pass. Sometimes we should play like brain games here is the one that is hangman. The brain games are which will help you in memory, response time, logic skills, problem solving etc.

**References:**

[https://www.python.org/](%20https:/www.python.org/)