REPORT FOR HANGMAN GAME

As a project work for Course

PYTHON PROGRAMMING (INT 213)

Name: Pratham Mishra

Roll No.:RK20YKB73

Registration No.: 12014732

Name: Pulkit Tikmani

Roll No.:RK20YKA32

Registration No.: 12013231

Program: **CSE B.Tech**

Semester: **3rd**

School: **School of Computer Science and Engineering**

Name of the University: **Lovely Professional University**

Date of Submission: **8th Nov2021**

Acknowledgement

I would like to express my sincere thanks and gratitude to my subject teacher **P.Raja Sir** for letting me work on this project **Hangman Game**. I am very grateful to him for his support and guidance in completing this project. His readiness for consultation at all times, his educative comments, his concern and assistance even with practical things have been invaluable.

I am thankful to my friends and seniors as well. I was able to successfully complete this project with the help of their guidance and support without whom it would not have been possible for me to understand and complete the project.

Table of Content

ORGANIZATION OF THE REPORT

The report is divided into various chapters and is organized as follows:

Chapter 1: Introduction

This chapter includes brief introduction to Python Programming Language and its applications.

Chapter 2: System requirements

This chapter includes details of hardware and software requirements necessary for theexecution of the project.

Chapter 3: Implementation and Results

This chapter includes the program code of the project and the results of successful runs of the code.

Conclusion

This section includes the conclusion about the project.

References

This section includes the bibliographical references used for the development of the project.

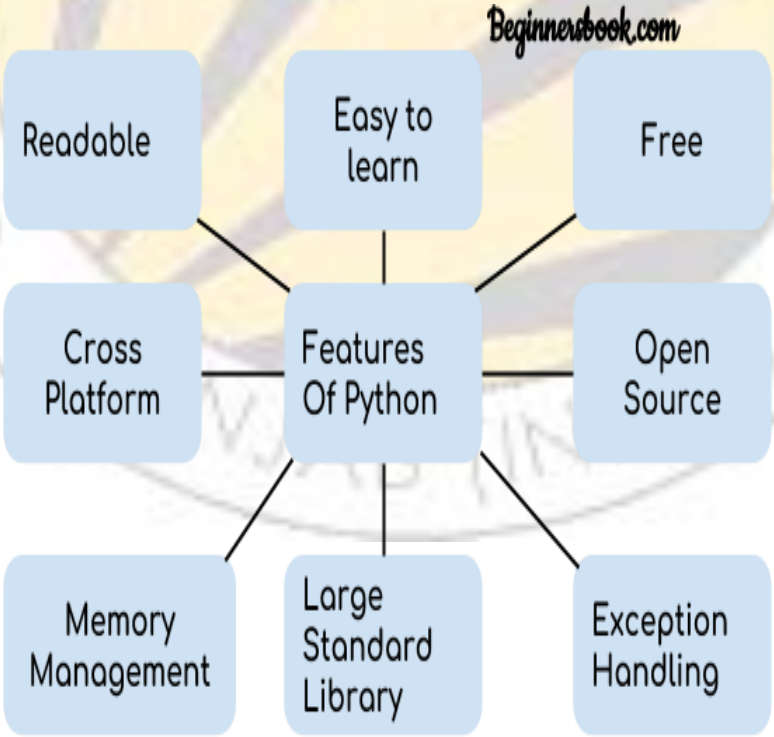
CHAPTER 1

INTRODUCTION

1.1 PYTHON PROGRAMMING LANGUAGE:

Python is one of the many open source object oriented programming application software available in the market . Python is developed by **Guido van Rossum**. Guido van Rossum started implementing Python in 1989. Python is a very simple programming language so even if you are new to programming, you can learn python without facing any issues. Some of the many uses of Python are application development, implementation of automation testing process, allows multiple programming build, fully constructed programming library, can be used in all the major operating systems and platforms, database system accessibility, simpleand readable code, easy to apply on complex software development processes, aids in test driven software application development approach, machine learning/ data analytics, helps pattern recognitions, supported in multiple tools, permitted by many of the provisioned frameworks, etc.

Some features of Python are



1. **Readable:** Python is a very readable language.

2. **Easy to Learn:** Learning python is easy as this is a expressive and high level programminglanguage, which means it is easy to understand the language and thus easy to learn.

3. **Cross platform:** Python is available and can run on various operating systems such as Mac,Windows, Linux, Unix etc. This makes it a cross platform and portable language.

4. **Open Source:** Python is a open source programming language.

5. **Large standard library:** Python comes with a large standard library that has some handy codes and functions which we can use while writing code in Python.

6. **Free:** Python is free to download and use. This means you can download it for free and use it inyour application. See: Open Source Python License. Python is an example of a FLOSS (Free/Libre Open Source Software), which means you can freely distribute copies of this software, read its source code and modify it.

7. **Supports exception handling:** If you are new, you may wonder what is an exception? An exception is an event that can occur during program exception and can disrupt the normal flow of program. Python supports exception handling which means we can write less error prone code and can test various scenarios that can cause an exception later on.

8. **Advanced features:** Supports generators and list comprehensions. We will cover these features later.

9. **Automatic memory management:** Python supports automatic memory management which means the memory is cleared and freed automatically.

1.2 Applications of Python programming language

Python can be used to develop different applications like web applications, graphic user interface based applications, software development application, scientific and numeric applications, network programming, Games and 3D applications and other business applications. It makes an interactive interface and easy development of applications.You may be wondering what all are the applications ofPython. There are so many applications of Python, here are some of the them.

1. ***Web development*** – Web framework like Django and Flask are based on Python. They help you write server side code which helps you manage database, write backend programming logic, mapping urls etc.
2. ***Machine learning*** – There are many machine learning applications written in Python. Machine learning is a way to write a logic so that a machine can learn and solve a particular problem on its own. For example, products recommendation in websites like Amazon, Flipkart, eBay etc. is amachine learning algorithm that recognizes user's interest. Face recognition and Voice recognition in your phone is another example of machine learning.
3. ***Data Analysis*** – Data analysis and data visualization in form of charts can also be developed using Python.
4. ***Scripting*** – Scripting is writing small programs to automate simple tasks such as sending automatedresponse emails etc. Such type of applications can also be written in Python programming language.
5. ***Game development*** – You can develop games using Python.
6. You can develop ***Embedded applications*** in Python.
7. ***Desktop applications*** – You can develop desktop application in Python using library like TKinter orQT.

CHAPTER 2

SOFTWARE REQUIREMENTS

2.1 SOFTWARE REQUIREMENTS

PyCharm2021.2.3(Community Edition)

Tkinter Directory

2.2 HARDWARE REQUIREMENTS

Operating System: WINDOWS 11

PROCESSOR : intel core I7

DISK SPACE : 512 SSD.

CHAPTER 3

IPLEMENTATION AND RESULTS

3.1 About Project

This is a simple Hangman game using Python programming language. Beginners can use this as asmall 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 the chances are over .
4. For convenience, we have given length of word +2 chances . For example , word to be guessed is mango , then user gets 5 + 2 = 7 chances , as mango is a five letter word.

3.2 Project Code

Snapshot of Project CODE

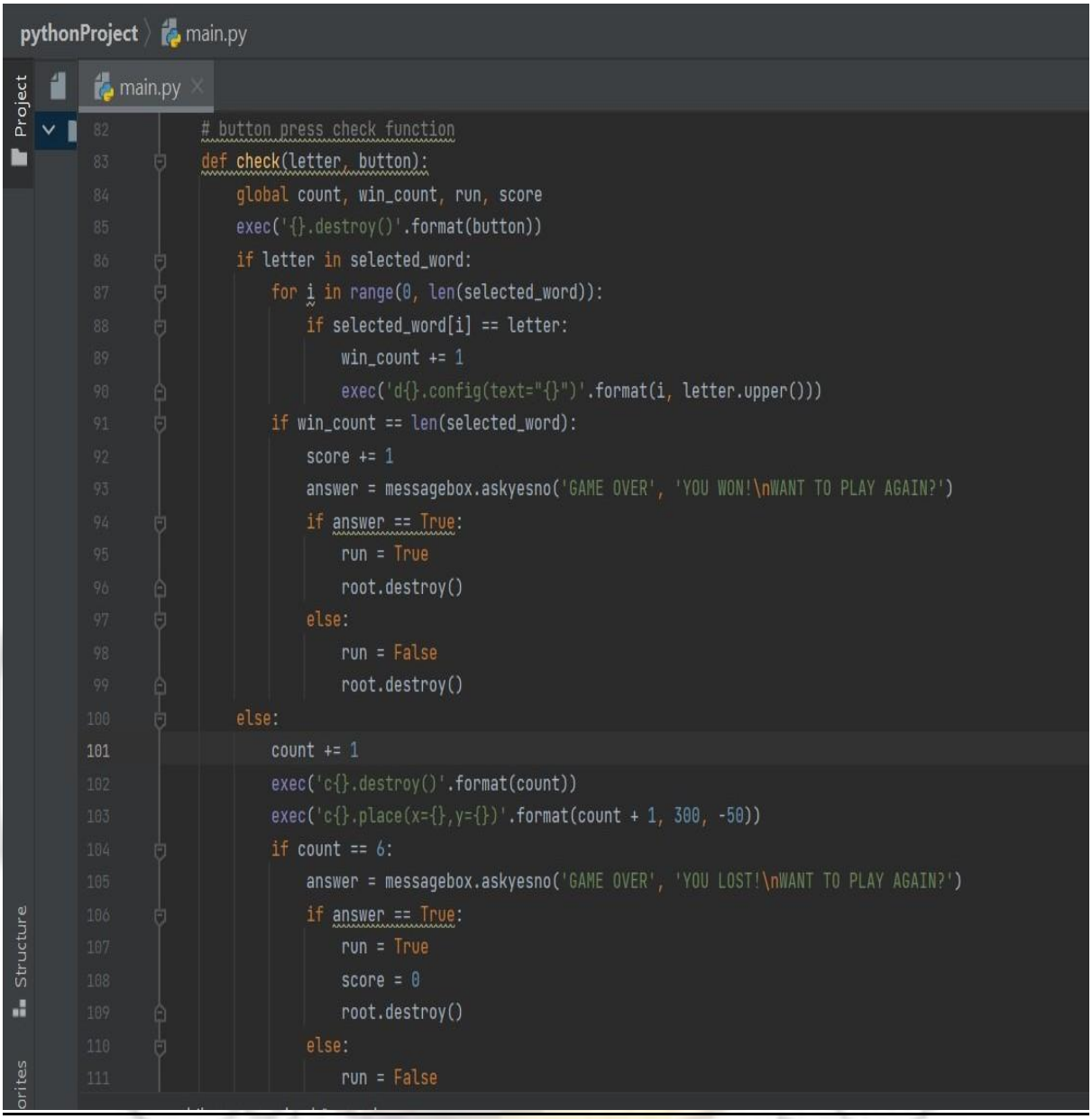
main.py

```
import random
from tkinter import *
from tkinter import messagebox

score = 0
run = True

# main loop
while run:
    root = Tk()
    root.geometry('900x900')
    root.title('HANG MAN')
    root.config(bg='#EADEDE')
    count = 0
    win_count = 0

    # choosing word
    index = random.randint(0, 853)
    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="#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']
```

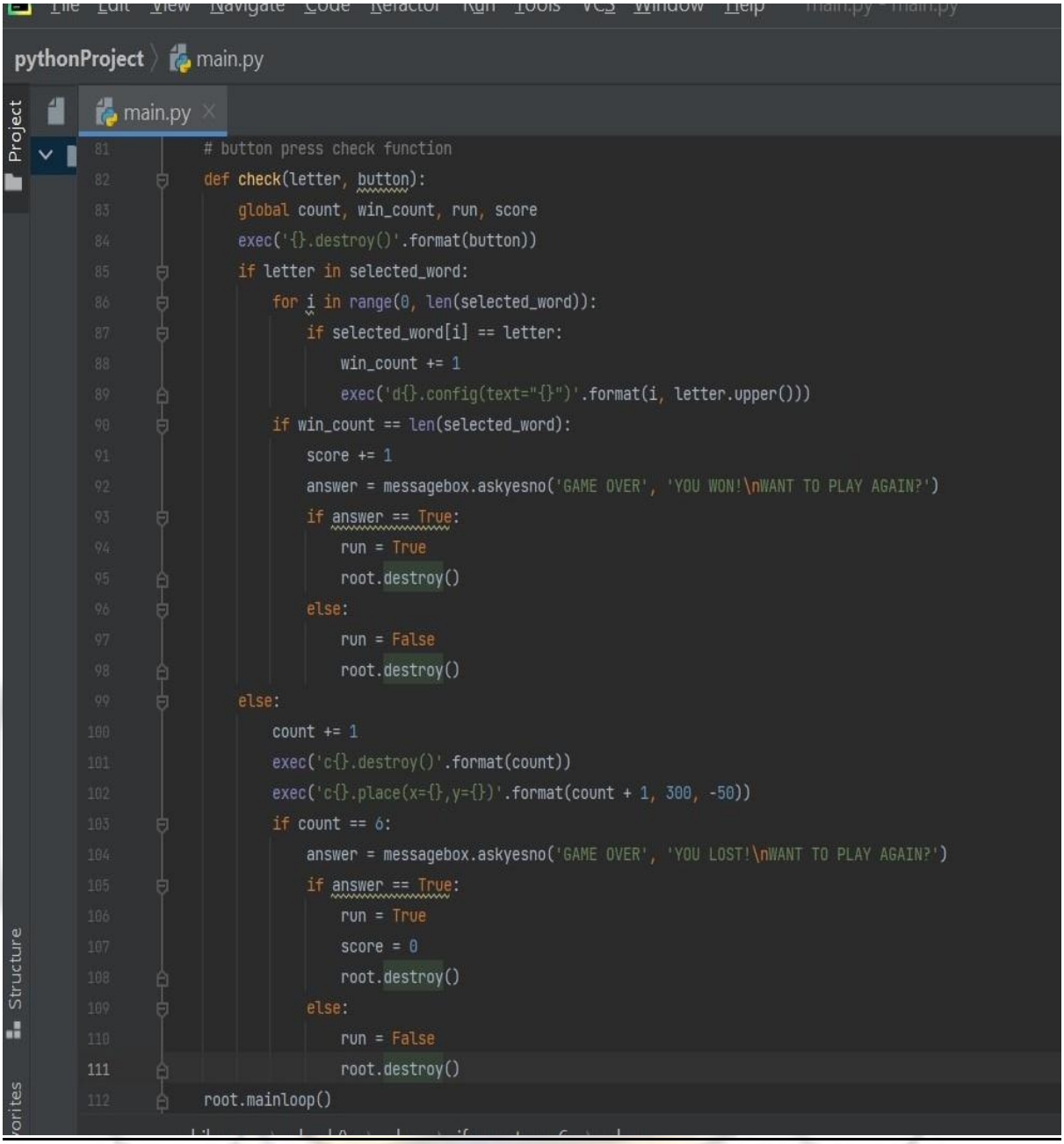

main.py

```
    # 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="#EADEDE",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))
```

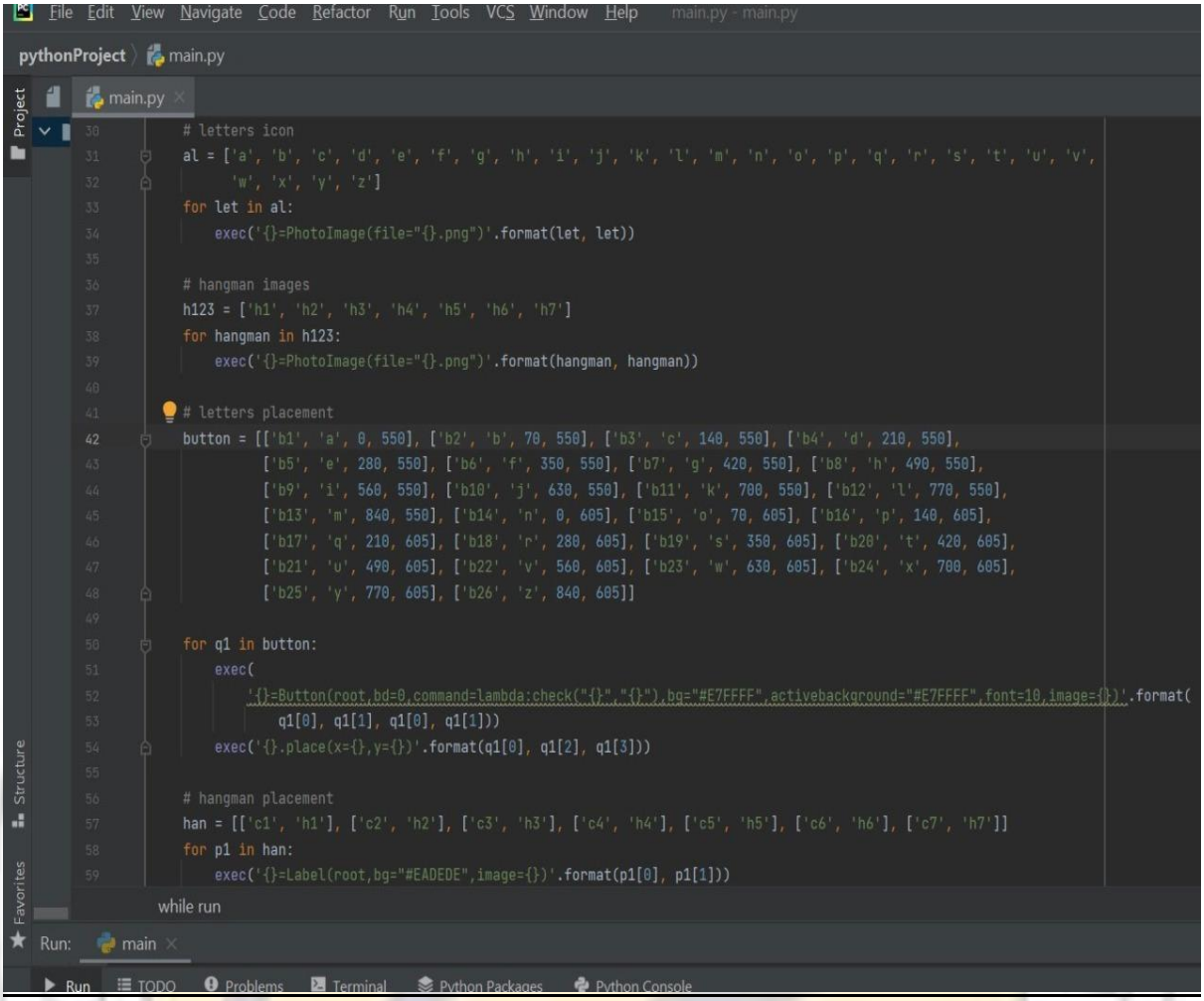
while run



```
pythonProject > main.py

main.py

# 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
```



```
# 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()
```




```
# 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, 550], ['b2', 'b', 70, 550], ['b3', 'c', 140, 550], ['b4', 'd', 210, 550],
          ['b5', 'e', 280, 550], ['b6', 'f', 350, 550], ['b7', 'g', 420, 550], ['b8', 'h', 490, 550],
          ['b9', 'i', 560, 550], ['b10', 'j', 630, 550], ['b11', 'k', 700, 550], ['b12', 'l', 770, 550],
          ['b13', 'm', 840, 550], ['b14', 'n', 0, 605], ['b15', 'o', 70, 605], ['b16', 'p', 140, 605],
          ['b17', 'q', 210, 605], ['b18', 'r', 280, 605], ['b19', 's', 350, 605], ['b20', 't', 420, 605],
          ['b21', 'u', 490, 605], ['b22', 'v', 560, 605], ['b23', 'w', 630, 605], ['b24', 'x', 700, 605],
          ['b25', 'y', 770, 605], ['b26', 'z', 840, 605]]

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="#EADEDE",image={})'.format(p1[0], p1[1]))

while run
```

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.