CHAPTER 1

INTRODUCTION

1.1 Introduction

The Game Zone is the simple project in python. This Game Zone is a gaming application with multiple games. As this gaming application consists of multiple games, the user has to select the game that he wants to play. In this gaming app, the player has to use keyboard or mouse for selecting and playing the games. There is a simple and clean GUI (tkinter) for easy game play. It includes both multiplayer and computerized games.

1.2 Objectives

The Game Zone app was built using Python module. The game is played in a simple windowed interface that displays appropriate options to control the game. The player can play the game using either keyboard or mouse. This Game Zone app consists of three mini games, Tic-tac-toe, Hangman and Rock-Paper-Scissor.

1.3 System Overview

Here is a high-level overview of how a Game Zone is structured:

Setting up the GUI: The game's GUI is typically created using tkinter. This involves setting up a window, defining the game's graphics, and creating any necessary buttons or menus.
Defining the game logic: The game logic includes defining how the player can interact with the game, how collisions are detected, and how the game.

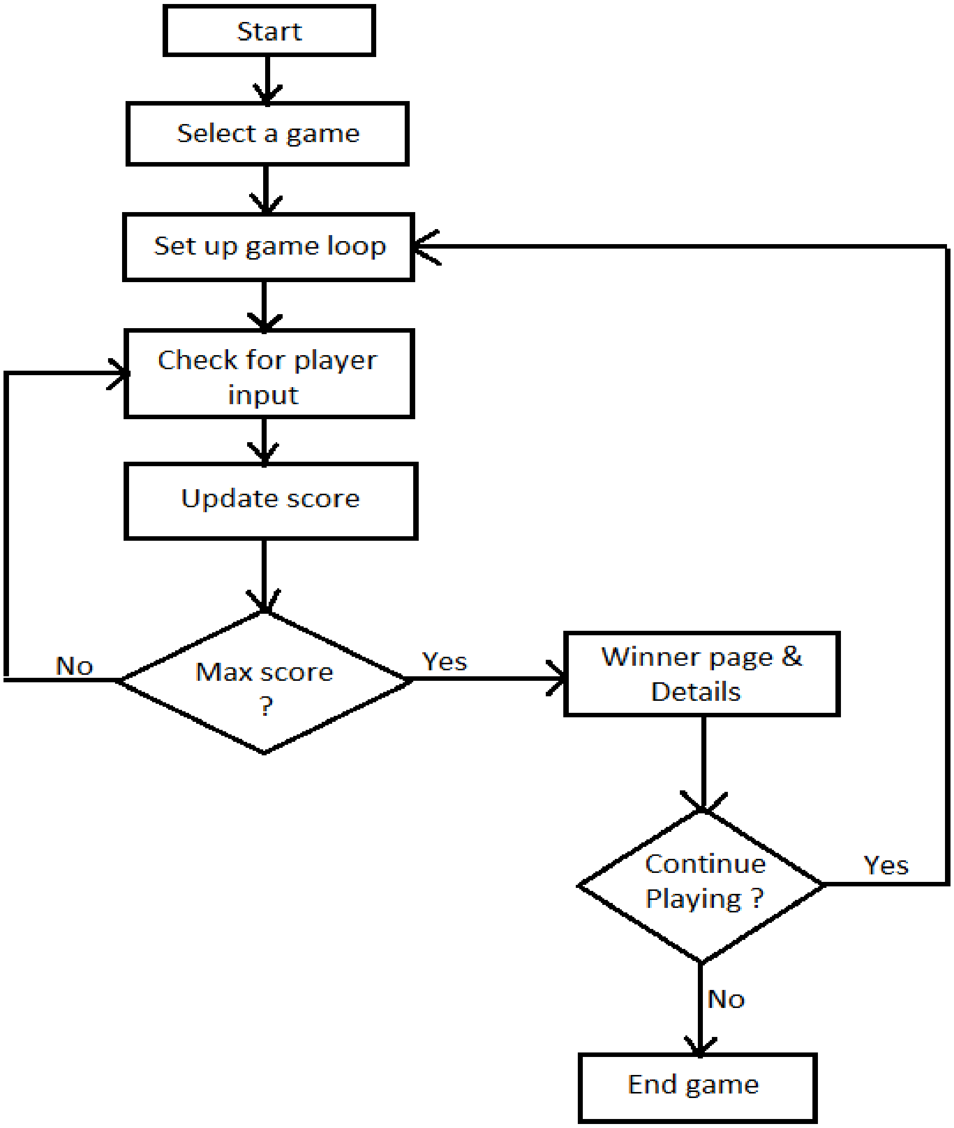


Figure-1.1: Flow diagram of Game Zone.

CHAPTER 2

METHODOLOGY

2.1 Classification

Set up the game window: You can use a Python module like tkinter to create a game window with a certain resolution and colour scheme.

Create the game objects: You need to create objects like buttons, score baord and other elements that will appear on the screen. These objects can be represented as images or shapes, depending on the style of the game.

Add scoring: You can keep track of the player's score . The score can be incremented as the player movement, and can be displayed on the screen.

Add game over screen: When the game ends, you can display a game over screen with the final score and an option to restart the game.

2.2 Features:

- Multiple Games : It includes multiple games with options like Multiplayer and single player games.
- Game window: The game should have a window where the player can see the game environment.
- Score: The game should keep track of the player's score, which increases as the player survives longer.
- High score: The game can have a feature where the player's high score is saved and displayed at the end of each game.
- Game over: The game should end when the player collides with an obstacle or goes out of bounds**.**

CHAPTER 3

IMPLEMENTATION

3.1 Graphical User Interface (GUI)

The graphical user interface has all the options needed for the software. The software serves administration and other debugging purposes. We don't need to edit code for any management. For example, if we need to open any game, we can do by clicking the button.

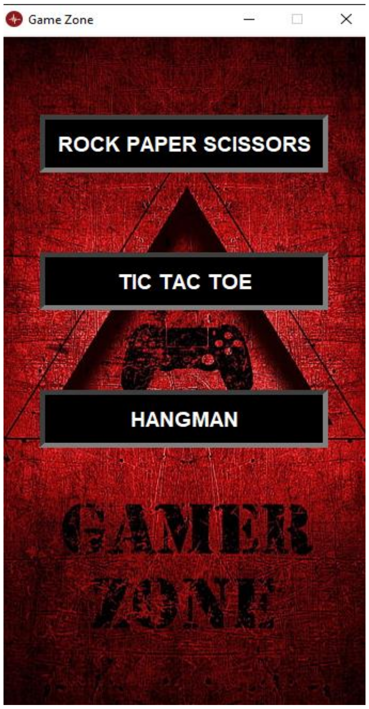
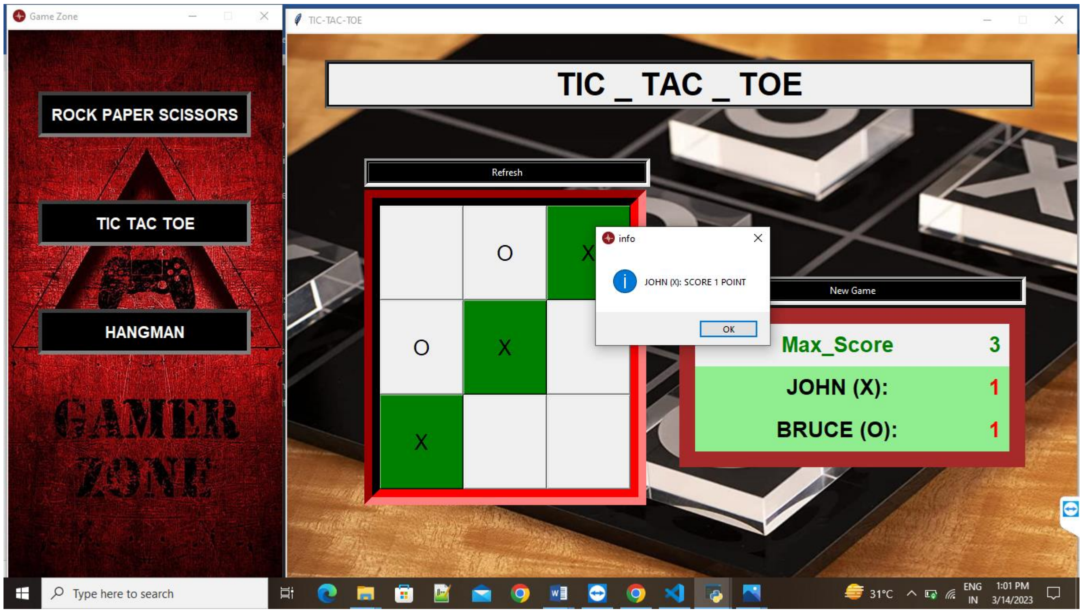


Figure 3.1: Open Game Zone

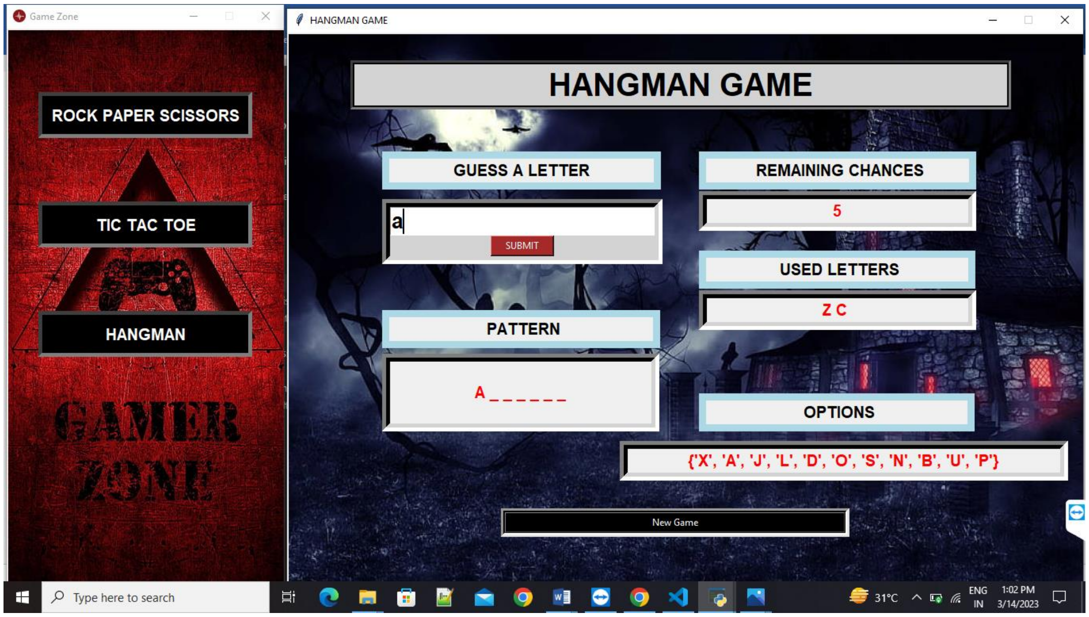


	O	X
O	X	
X		

JOHN (X): SCORE 1 POINT

Max_Score	3
JOHN (X):	1
BRUCE (O):	1

Figure 3.2: GUI of Tic_Tac_Toe Game.



GUESS A LETTER: a

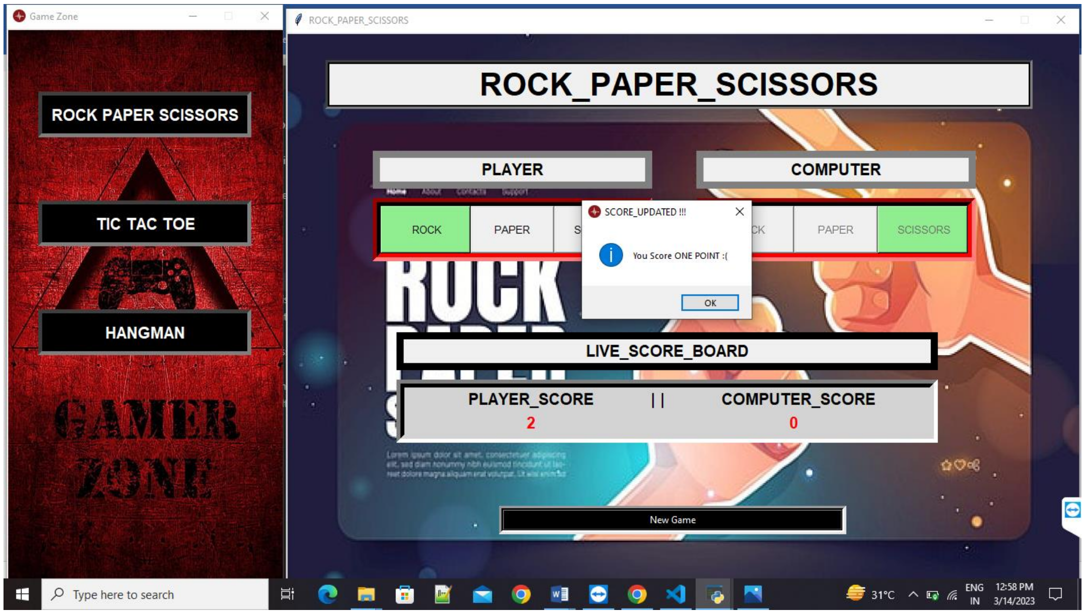
REMAINING CHANCES: 5

USED LETTERS: Z C

PATTERN: A _ _ _ _ _ _

OPTIONS: {'X', 'A', 'J', 'L', 'D', 'O', 'S', 'N', 'B', 'U', 'P'}

Figure 3.3: GUI of Hangman Game.



SCORE_UPDATED !!!

You Score ONE POINT :(

PLAYER_SCORE || COMPUTER_SCORE
2 0

Figure 3.4: GUI of Rock_Paper_Scissor Game.

3.2 Source code:

#main.py

```
from tkinter import *
from hangman_game_gui import *
from rock_paper_scissors_gui import *
from tic_tac_toe_multiplayer import *
t=Tk()
t.title('Game Zone')
t.geometry('350x729+0+5')
t.resizable(0,0)
t.wm_iconbitmap('life.ico')
t['bg']='steelblue4'

bg=PhotoImage(file='img1.png')
```

```
img=Label(t,image=bg)
img.place(x=0,y=0)

b1=Button(t,text="ROCK PAPER SCISSORS",command=rockPaperScissors,width=200,height=1,font=('ariel',15,'bold'),relief=SUNKEN,bd=10,fg='white',bg='black')
b1.pack(padx=40,pady=80)
b2=Button(t,text="TIC TAC TOE",command=ticTacToe,width=200,height=1,font=('ariel',15,'bold'),relief=SUNKEN,bd=10,fg='white',bg='black')
b2.pack(padx=40,pady=0)
b3=Button(t,text="HANGMAN",command=hangman,width=200,height=1,font=('ariel',15,'bold'),relief=SUNKEN,bd=10,fg='white',bg='black')
b3.pack(padx=40,pady=80)
t.mainloop()

#Tic-tac-toe.py

from tkinter import *
from tkinter import messagebox as m

def ticTacToe():
    global clicked
    rt  = Toplevel()
    rt.title('TIC-TAC-TOE')
    rt.geometry('1005x729+350+5')
    rt['bg'] = 'lightblue'
    rt.resizable(0,0)

    bg=PhotoImage(file='t2.png')
    img=Label(rt,image=bg)
    img.place(x=0,y=0)
```

```
clicked = True
count = 0
def disable_buttons():
    l=[b1,b2,b3,b4,b5,b6,b7,b8,b9]
    for i in l:
        i.config(state=DISABLED)
def won():
    if b1['text'] == b2['text'] == b3['text'] != ' ':
        b1.config(bg = 'green')
        b2.config(bg = 'green')
        b3.config(bg = 'green')
        xwin() if b1['text'] == b2['text'] == b3['text'] == 'X' else owin()
    elif b4['text'] == b5['text'] == b6['text'] != ' ':
        b4.config(bg = 'green')
        b5.config(bg = 'green')
        b6.config(bg = 'green')
        xwin() if b4['text'] == b5['text'] == b6['text'] == 'X' else owin()
    elif b7['text'] == b8['text'] == b9['text'] != ' ':
        b7.config(bg = 'green')
        b8.config(bg = 'green')
        b9.config(bg = 'green')
        xwin() if b7['text'] == b8['text'] == b9['text'] == 'X' else owin()
    elif b1['text'] == b4['text'] == b7['text'] != ' ':
        b1.config(bg = 'green')
        b4.config(bg = 'green')
        b7.config(bg = 'green')
        xwin() if b1['text'] == b4['text'] == b7['text'] == 'X' else owin()
    elif b2['text'] == b5['text'] == b8['text'] != ' ':
        b2.config(bg = 'green')
        b5.config(bg = 'green')
        b8.config(bg = 'green')
        xwin() if b2['text'] == b5['text'] == b8['text'] == 'X' else owin()
    elif b3['text'] == b6['text'] == b9['text'] != ' ':
        b3.config(bg = 'green')
        b6.config(bg = 'green')
        b9.config(bg = 'green')
        xwin() if b3['text'] == b6['text'] == b9['text'] == 'X' else owin()
    elif b1['text'] == b5['text'] == b9['text'] != ' ':
        b1.config(bg = 'green')
        b5.config(bg = 'green')
        b9.config(bg = 'green')
        xwin() if b1['text'] == b5['text'] == b9['text'] == 'X' else owin()
    elif b3['text'] == b5['text'] == b7['text'] != ' ':
        b3.config(bg = 'green')
        b5.config(bg = 'green')
        b7.config(bg = 'green')
        xwin() if b3['text'] == b5['text'] == b7['text'] == 'X' else owin()
    elif count == 9:
        m.showinfo('Oops!!','It\'s a TIE')
        refresh()
def owin():
    c_inp = lblo.cget('text')
    op = int(c_inp)+1
    lblo.config(text=str(op))
    o_name = lbl2.cget('text')
    winner(o_name,op,'O')
    if op != 3: m.showinfo('info',f'{o_name} SCORE 1 POINT')
    refresh()
def xwin():
    c_inp = lblx.cget('text')
    op = int(c_inp)+1
    lblx.config(text=str(op))
    x_name = lbl1.cget('text')
    winner(x_name,op,'X')
    if op != 3: m.showinfo('info',f'{x_name} SCORE 1 POINT')
    refresh()
def winner(name,op,id):
    if op == 3:
        rt.config(bg="lightgreen")
        if id=='X':
            lblx.config(bg='green',fg='white')
            lbl1.config(bg='green')
            lblo.config(bg='red',fg='white')
            lbl2.config(bg='red')
        else:
            lblx.config(bg='red',fg='white')
            lbl1.config(bg='red')
            lblo.config(bg='green',fg='white')
            lbl2.config(bg='green')
        m.showinfo('GAME OVER !!!',f' {name} is the winner!!!')
        newGame()
def b_click(b):
    global clicked,count
    if b['text'] == ' ' and clicked == True:
        b['text'] = 'X'
        clicked = False
        count += 1
        won()
    elif b['text'] == ' ' and clicked == False:
        b['text'] = 'O'
        clicked = True
        count += 1
        won()
    else:
        m.showerror('tic-tac-toe','Its not empty !!!!')
def refresh():
    global count
    count = 0
    b=[b1,b2,b3,b4,b5,b6,b7,b8,b9]
    for i in b:
        i.config(text=' ')
        i.config(state=ACTIVE)
        i.config(fg='black')
        i.config(bg='SystemButtonFace')
def newGame():
    refresh()
    lbl1.config(bg='lightgreen')
    lbl2.config(bg='lightgreen')
    lblo.config(text='0',bg='lightgreen',fg='red')
    lblx.config(text='0',bg='lightgreen',fg='red')
    rt.config(bg="lightblue")

    y = m.askyesno('NEW GAME',"Woul\'d You Like To Play Again ???")
    if y == 1:
        ask_to_change_name()
    else:
        rt.withdraw()

def ask_to_change_name():
    x = m.askyesno('player_name','woul\'d you like to change the Name of Players')
    if x == 1:
        change_name()

def change_name():
    def save_name():
        new_x = p_entry1.get()
        new_o = p_entry2.get()
        lbl1.config(text = new_x.upper()+' (X):')
```

```
            lbl2.config(text = new_o.upper()+' (O):')
            t.withdraw()
        global p_entry1,p_entry2
        t=Tk()
        t.geometry('690x150+550+200')
        t.title('Change Player_Names')
        top1 = Frame(t,bg = 'lightgreen',bd=20)
        top1.pack(side=TOP)
        player1 = Label(top1,text="player 'X'  :",font=('aria',20,'bold'),width=20)
        player2 = Label(top1,text="player 'O'  :",font=('aria',20,'bold'),width=20)
        p_entry1 = Entry(top1,font=('aria',20,'bold'),width=20)
        p_entry2 = Entry(top1,font=('aria',20,'bold'),width=20)
        btn = Button(top1,text='SAVE',command =
save_name,font=('aria',10,'bold'),width=10,relief=SUNKEN,bg='red',fg='white')
        player1.grid(row=1,column=1)
        player2.grid(row=2,column=1)
        p_entry1.grid(row=1,column=2)
        p_entry2.grid(row=2,column=2)
        btn.grid(row=3,column=2)
        t.mainloop()

    top = Frame(rt,bg="black",width=800,height=50,relief=SUNKEN,bd=5)
    top.place(x=50,y=35)
    player = Frame(rt,bg="brown",width=500,height=500,bd=20)
    player.place(x=500,y=350)
    newgame_btn = Frame(rt,bg="white",width=500,height=500,relief=SUNKEN,bd=5)
    newgame_btn.place(x=500,y=310)
    game = Frame(rt,bg="red",width=500,height=500,bd=20,relief=SUNKEN)
    game.place(x=100,y=200)
    refresh_btn = Frame(rt,bg="white",width=500,height=500,relief=SUNKEN,bd=5)
    refresh_btn.place(x=100,y=160)
```

```
   reset_btn = Button(newgame_btn,text = 'New Game',command = lambda
:newGame(),width=60,bg='black',fg='white')
   reset_btn.grid(row=5,column=0)
   reset_btn = Button(refresh_btn,text = 'Refresh',command = lambda
:refresh(),width=49,bg='black',fg='white')
   reset_btn.grid(row=5,column=1)

   lbl_max= Label(player,text="Max_Score",font=('aria',20,'bold'),fg='green',width=20,bd=10)
   lbl_max_score = Label(player,text='3',font=('aria',20,'bold'),fg= 'green',bd=10)
   lbl1 = Label(player,text="player 'X'  : ",font=('aria',20,'bold'),width=20,bd=10,bg='lightgreen')
   lblx = Label(player,text='0',font=('aria',20,'bold'),fg= 'red',bd=10,bg='lightgreen')
   lbl2 = Label(player,text="player 'O'  : ",font=('aria',20,'bold'),width=20,bd=10,bg='lightgreen')
   lblo = Label(player,text='0',font=('aria',20,'bold'),fg= 'red',bd=10,bg='lightgreen')
   lbl_max.grid(row=1,column=0)
   lbl_max_score.grid(row=1,column=1)
   lbl1.grid(row=2,column=0)
   lbl2.grid(row=3,column=0)
   lblx.grid(row=2,column=1)
   lblo.grid(row=3,column=1)

   head=Label(top,font=('aria',30,'bold'),text="                    TIC _ TAC _ TOE
")
   head.grid(row=2,column=0)

   b1=Button(game,text=" ",font=("Helvetica",20),height=3,width=6,bg='SystemButtonFace',
command =lambda : b_click(b1))
   b2=Button(game,text=" ",font=("Helvetica",20),height=3,width=6,bg='SystemButtonFace',
command =lambda : b_click(b2))
   b3=Button(game,text=" ",font=("Helvetica",20),height=3,width=6,bg='SystemButtonFace',
command =lambda : b_click(b3))

   b4=Button(game,text=" ",font=("Helvetica",20),height=3,width=6,bg='SystemButtonFace',
command =lambda : b_click(b4))
```

```
    b5=Button(game,text=" ",font=("Helvetica",20),height=3,width=6,bg='SystemButtonFace', command =lambda : b_click(b5))
    b6=Button(game,text=" ",font=("Helvetica",20),height=3,width=6,bg='SystemButtonFace', command =lambda : b_click(b6))

    b7=Button(game,text=" ",font=("Helvetica",20),height=3,width=6,bg='SystemButtonFace', command =lambda : b_click(b7))
    b8=Button(game,text=" ",font=("Helvetica",20),height=3,width=6,bg='SystemButtonFace', command =lambda : b_click(b8))
    b9=Button(game,text=" ",font=("Helvetica",20),height=3,width=6,bg='SystemButtonFace', command =lambda : b_click(b9))

    b1.grid(row=1,column=0)
    b2.grid(row=1,column=1)
    b3.grid(row=1,column=2)

    b4.grid(row=2,column=0)
    b5.grid(row=2,column=1)
    b6.grid(row=2,column=2)

    b7.grid(row=3,column=0)
    b8.grid(row=3,column=1)
    b9.grid(row=3,column=2)
    ask_to_change_name()
    rt.mainloop()
     #Hangman

     import random

     from tkinter import *

     from tkinter import messagebox as m

     from word_list import word_list1 as word

     from word_list import a_z
```

```
def hangman():
    rt  = Toplevel()
    rt.title('HANGMAN GAME')
    rt.geometry('1005x729+350+5')
    rt['bg'] = 'lightblue'
    rt.resizable(0,0)

    bg=PhotoImage(file='h2.png')
    img=Label(rt,image=bg)
    img.place(x=0,y=0)

    top = Frame(rt,bg="black",width=800,height=50,relief=SUNKEN,bd=5)
    top.place(x=80,y=35)
    head=Label(top,font=('aria',30,'bold'),text="                    HANGMAN GAME
",bg='lightgray')
    head.grid(row=2,column=0)

    #Guess A Letter
    guess = Frame(rt,bg="lightblue",width=350,height=150,bd=9)
    guess.place(x=120,y=150)
    head2=Label(guess,font=('aria',15,'bold'),text="            GUESS A LETTER            ")
    head2.grid(row=1,column=0)
    screen = Frame(rt,bg="lightgray",width=350,height=100,bd=10,relief=SUNKEN)
    screen.place(x=120,y=210)


    screen_in = Entry(screen,width=22,font=('aria',20,'bold'))
    screen_in.pack()
    submit_btn = Button(screen,text = 'SUBMIT',command = lambda :sub(),width=10,bg='brown',fg='white')
    submit_btn.pack()
    #Pattern
    pattern1 = Frame(rt,bg="lightblue",width=350,height=150,bd=9)
    pattern1.place(x=120,y=350)
    head1=Label(pattern1,font=('aria',15,'bold'),text="                PATTERN                ")
    head1.grid(row=1,column=0)
    dis1 = Frame(rt,bg="lightgray",width=350,height=100,bd=10,relief=SUNKEN)
    dis1.place(x=120,y=405)
    lbl1=Label(dis1,font=('aria',15,'bold'),text="",fg='red',width=27,height=3)
    lbl1.grid(row=1,column=0)
    #Chances Remaining
    chances1 = Frame(rt,bg="lightblue",width=350,height=150,bd=9)
    chances1.place(x=520,y=150)
    head2=Label(chances1,font=('aria',15,'bold'),text="        REMAINING CHANCES        ")
    head2.grid(row=1,column=0)
    dis2 = Frame(rt,bg="lightgray",width=350,height=100,bd=10,relief=SUNKEN)
```

```
dis2.place(x=520,y=200)
lbl2=Label(dis2,font=('aria',15,'bold'),text="0",fg='red',width=27,height=1)
lbl2.grid(row=1,column=0)

#Trash/usedletters
trash1 = Frame(rt,bg="lightblue",width=350,height=150,bd=9)
trash1.place(x=520,y=275)
head3=Label(trash1,font=('aria',15,'bold'),text="            USED LETTERS            ")
head3.grid(row=1,column=0)
dis3 = Frame(rt,bg="lightgray",width=350,height=100,bd=10,relief=SUNKEN)
dis3.place(x=520,y=325)
lbl3=Label(dis3,font=('aria',15,'bold'),text="",fg='red',width=27,height=1)
lbl3.grid(row=1,column=0)
#Options
trash1 = Frame(rt,bg="lightblue",width=350,height=150,bd=9)
trash1.place(x=520,y=455)
head3=Label(trash1,font=('aria',15,'bold'),text="                OPTIONS                ")
head3.grid(row=1,column=0)
dis4 = Frame(rt,bg="lightgray",width=350,height=100,bd=10,relief=SUNKEN)
dis4.place(x=420,y=515)
lbl4=Label(dis4,font=('aria',15,'bold'),text="",fg='red',width=45,height=1)
lbl4.grid(row=1,column=0)
```

```
newgame = Frame(rt,bg="white",width=500,height=500,relief=SUNKEN,bd=5)

newgame.place(x=270,y=600)

reset_btn = Button(newgame,text = 'New Game',command = lambda
:newGame(),width=60,bg='black',fg='white')

reset_btn.grid(row=5,column=0)

#functions

def newGame():

    head.config(bg='lightgray',fg='black')

    global chances,pattern,trash,new_word,new_word_prs

    new_word = random.choice(word).upper()

    new_word_prs = list(new_word)

    chances = len(new_word)

    pattern = list(chances*'_')

    trash = ''

    opt = new_word

    for _ in range(chances):

        opt += random.choice(a_z).upper()

    options = set(opt)

    lbl2.config(text=str(chances))

    lbl1.config(text=str(' '.join(pattern)))

    lbl4.config(text=str(options))

    lbl3.config(text=str(trash))
    screen_in.delete(0,END)
def askToPlayAgain():
    y = m.askyesno('NEW GAME',"Woul\'d You Like To Play Again ???")
    if y == 1:
        newGame()
        return 1
    else:
        rt.withdraw()
def trashF():
    global chances,pattern,trash,new_word,new_word_prs
    char = screen_in.get().upper()
    trash+=char
    trash+=' '
    lbl3.config(text=str(trash))#trash updating
    screen_in.delete(0,END)
    chances-=1
    lbl2.config(text=str(chances))
def sub():
        global chances,pattern,trash,new_word,new_word_prs

        char = screen_in.get().upper()
        if len(char) > 1 and char != new_word:
```

```
        trashF()
    if char == new_word :
            lbl1.config(text=str(' '.join(list(new_word))))#pattern updatting
            head.config(bg='green',fg='white')
            m.showinfo('** G O T C H A **','YOU WON, \n YOU SAVED HIM ##')
            askToPlayAgain()
    elif char.isalpha():
        if len(char)==1 :
            if char in new_word_prs:
                ind = new_word_prs.index(char)
                pattern[ind]=char
                new_word_prs[ind] = '_'
                screen_in.delete(0,END)
                lbl1.config(text=str(' '.join(pattern)))#pattern updatting
            else:
                trashF()
        else:
            screen_in.delete(0,END)
            m.askretrycancel("CHAR COUNT","You Have To Enter A SINGLE CHARACTER !!")
    else:
        m.askretrycancel("NOT AN ALPHABET","You Have To Enter An ALPHABET !!")
    if chances < 1:
```

```
             head.config(bg='red',fg='white')

             m.showerror('G A M E  O V E R !!',' Out Of Chances,\n YOU KILLED HIM !!!')

             askToPlayAgain()

          if '_' not in pattern:

             head.config(bg='green',fg='white')

             m.showinfo('** G O T C H A **','YOU WON, \n YOU SAVED HIM ##')

             askToPlayAgain()


    newGame()

    rt.mainloop()


  # Rock-Paper-Scissor.py

  import random

 from tkinter import *

 from tkinter import messagebox as m


def rockPaperScissors():

 rt  = Toplevel()

  rt.title('ROCK_PAPER_SCISSORS')

  rt.geometry('1005x729+350+5')

  rt['bg'] = 'lightblue'

  rt.resizable(0,0)
```

```
bg=PhotoImage(file='r1.png')

img=Label(rt,image=bg)

img.place(x=0,y=0)



p_score = 0

m_score = 0

n = 1



top = Frame(rt,bg="black",width=800,height=50,relief=SUNKEN,bd=5)

top.place(x=50,y=35)

head=Label(top,font=('aria',30,'bold'),text="ROCK_PAPER_SCISSORS")

head.grid(row=2,column=0)

#player

player = Frame(rt,bg="gray",width=350,height=150,bd=9)

player.place(x=110,y=150)

head1=Label(player,font=('aria',15,'bold'),text="PLAYER")

head1.grid(row=1,column=0)

game1 = Frame(rt,bg="red",width=350,height=150,bd=10,relief=SUNKEN)

game1.place(x=110,y=210)

pb1=Button(game1,text="    ROCK
",font=("Helvetica",10),height=3,width=13,bg='SystemButtonFace', command =lambda : bClick('r'))
```

```
        pb2=Button(game1,text="   PAPER   ",font=("Helvetica",10),height=3,width=12,bg='SystemButtonFace', command =lambda : bClick('p'))

        pb3=Button(game1,text="  SCISSORS   ",font=("Helvetica",10),height=3,width=13,bg='SystemButtonFace', command =lambda :bClick('s'))

        pb1.grid(row=2,column=0)

        pb2.grid(row=2,column=1)

        pb3.grid(row=2,column=2)

        #computer

        comp = Frame(rt,bg="gray",width=350,height=150,bd=9)

        comp.place(x=520,y=150)

        head2=Label(comp,font=('aria',15,'bold'),text="                    COMPUTER                    ")

        head2.grid(row=1,column=0)

        game2 = Frame(rt,bg="red",width=350,height=150,bd=10,relief=SUNKEN)

        game2.place(x=520,y=210)

        cb1=Button(game2,text="    ROCK    ",font=("Helvetica",10),height=3,width=13,bg='SystemButtonFace', command =lambda : bClick('r'),state=DISABLED)

        cb2=Button(game2,text="   PAPER   ",font=("Helvetica",10),height=3,width=12,bg='SystemButtonFace', command =lambda : bClick('p'),state=DISABLED)

        cb3=Button(game2,text="  SCISSORS   ",font=("Helvetica",10),height=3,width=13,bg='SystemButtonFace', command =lambda : bClick('s'),state=DISABLED)

        cb1.grid(row=2,column=0)

        cb2.grid(row=2,column=1)

        cb3.grid(row=2,column=2)
        #score_board

        score_board = Frame(rt,bg="black",width=350,height=150,bd=9)

        score_board.place(x=140,y=380)

        head=Label(score_board,font=('aria',15,'bold'),text="
LIVE_SCORE_BOARD                                    ")

        head.grid(row=1,column=0)

        score_frame = Frame(rt,bg="lightgray",width=680,height=150,bd=10,relief=SUNKEN)

        score_frame.place(x=140,y=440)

        p_score=Label(score_frame,font=('aria',15,'bold'),text="PLAYER_SCORE
|",bg='lightgray')

        p_score.grid(row=1,column=0)

        p_score_value=Label(score_frame,font=('aria',15,'bold'),text="0",fg='red',bg='lightgray')

        p_score_value.grid(row=2,column=0)

        c_score=Label(score_frame,font=('aria',15,'bold'),text="COMPUTER_SCORE
",bg='lightgray')

        c_score.grid(row=1,column=1)

        c_score_value=Label(score_frame,font=('aria',15,'bold'),text="0",fg='red',bg='lightgray')

        c_score_value.grid(row=2,column=1)



        newgame = Frame(rt,bg="white",width=500,height=500,relief=SUNKEN,bd=5)

        newgame.place(x=270,y=600)

        reset_btn = Button(newgame,text = 'New Game',command = lambda
:refresh(),width=60,bg='black',fg='white')

        reset_btn.grid(row=5,column=0)
```

```
if __name__ == '__main__':

    None if m.askyesno("TIC TAC TOE"," THE MAXIMUM SCORE TO WIN THIS GAME IS # 3\n WOULD YOU LIKE TO CONTINUE ?") else rt.withdraw()

def bClick(p_choice):

    c_choice = random.choice(['r','p','s'])

    colorMe(p_choice,c_choice) ##color the selected boxes

    if p_choice==c_choice:

        m.showinfo("IT'S A TIE !!!",'Both computer and You choose the SAME OPTION !!!')

    else:

        score(p_choice,c_choice)

    unColorMe(p_choice,c_choice)

def colorMe(p1,c1):

    if p1=='r':

        pb1.config(bg='lightgreen')

    if p1=='p':

        pb2.config(bg='lightgreen')

    if p1=='s':

        pb3.config(bg='lightgreen')

    if c1=='r':

        cb1.config(bg='lightgreen')

    if c1=='p':

        cb2.config(bg='lightgreen')

    if c1=='s':
        cb3.config(bg='lightgreen')
def unColorMe(p1,c1):
    if p1=='r':
        pb1.config(bg='SystemButtonFace')
    if p1=='p':
        pb2.config(bg='SystemButtonFace')
    if p1=='s':
        pb3.config(bg='SystemButtonFace')
    if c1=='r':
        cb1.config(bg='SystemButtonFace')
    if c1=='p':
        cb2.config(bg='SystemButtonFace')
    if c1=='s':
        cb3.config(bg='SystemButtonFace')
def refresh():
    p_score_value.config(text="0")
    c_score_value.config(text="0")
    head1.config(bg="lightblue")
    head2.config(bg="lightblue")
def newGame():
    y = m.askyesno('NEW GAME',"Woul\'d You Like To Play Again ???")
    if y == 1:
        refresh()
    else:
        rt.withdraw()
def score(p,c):
    p_score = p_score_value.cget('text')
    c_score = c_score_value.cget('text')
    # r<p and p<s and s<r #player will score
    if (p == 'p' and c == 'r') or (p == 's' and c == 'p') or (p == 'r' and c == 's'):
        # player++
        p_incr = int(p_score)+1
        p_score_value.config(text=str(p_incr))
        if p_incr == 3:
            head1.config(bg="green")
            head2.config(bg="red")
            m.showinfo('CONGRATS !!!','***     YOU WON THE GAME     ***')
            newGame()
        else:
            m.showinfo('SCORE_UPDATED !!!',' You Score ONE POINT :(')
        return 1
    else:
        # computer++
        c_incr = int(c_score)+1
        c_score_value.config(text=str(c_incr))

        if c_incr == 3:

            head1.config(bg="red")

            head2.config(bg="green")

            m.showinfo('GAME OVER !!!',' YOU LOOSE IT !! BETTER LUCK NEXT TIME')

            newGame()

        else:

            m.showinfo('SCORE_UPDATED !!!',' I Score ONE POINT :)')

        return 0

  rt.mainloop()

     #wordlist.py

     import string,random

     word_list1 = 
["aback","abaft","abandoned","abashed","aberrant","abhorrent","abiding","abject","ablaze","absorbing","abstracted","absurd","abundant","abusive","accept","acceptable","accessible","accidental","account ","acrid","act","action","activity","actor","actually","adamant","adaptable"……]

     a_z = list(string.ascii_lowercase)
```

3.3 Libraries used for Graphical User Interface:

3.3.1 Tkinter:

Tkinter is the standard GUI library for Python. Python when combined with Tkinter providesa fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps-

- Import the Tkinter module.
- Create the GUI application main window.
- Add one or more of the above-mentioned widgets to the GUI application.
- Enter the main event loop to tke action against each event triggered by the user.

The tkinter package ("Tk interface") is the standard Python interface to the Tcl/Tk GUI toolkit. Both Tk and tkinter are available on most Unix platforms, including macOS, as well as on Windows systems.

Tkinter supports a range of Tcl/Tk versions, built either with or without thread support. The official Python binary release bundles Tcl/Tk 8.6 threaded. See the source code for the tkinter module for more information about supported versions.

Tkinter is not a thin wrapper, but adds a fair amount of its own logic to make the experience more pythonic. This documentation will concentrate on these additions and changes, and refer to the official Tcl/Tk documentation for details that are unchanged.

CONCLUSION

The project has been completed successfully as specified by the requirements. The implementation and testing has been one in a step-by-step manner. Each module has been developed and tested individually to obtain the required output in the desired form. The future improvements can be made in certain areas of project. There is scope for extending the project to incorporate more features by including more games with multiplayer options.

BIBLIOGRAPHY

References:

[1] YouTube

[2] Google

[3] GeeksforGeeks.org