Mini project

**Topic Name :**

Classic Tic-Tac-Toe Game in Python

**Theory :**

Tic-Tac-Toe game is an easy game which is mostly played among children and it also helps them to improve their concentration.

The objective of this tic-tac-toe game python project is to build a tic-tac-toe game so you can play it without wasting paper and improve your concentration. To build this game we use the tkinter module with the concept of python.

To play this game we require two players to play one is X and the other is O and both players play by putting their marks in empty squares.

To win the game players have to get 3 of her marks in a row (up, down, across, or diagonally).

Project Prerequisites

To build tic-tac-toe game using python we require tkinter module and basic concept of python

Tkinter modules is a standard graphical user interface used to render graphics.

Tkinter.messagebox used to display message box

To install tkinter modules we used pip install command on command prompt:

pip install tkinter

Project File Structure

These are the step to build Tic-Tac-Toe game using python :

* Import modules
* Initialize window
* Function to check result
* Function to check the winner
* Define labels and buttons

1. Import modules

from tkinter import \*

import tkinter.messagebox as msg

In this step, we import tkinter and messsagebox module

2. Initialize window

root= Tk()

root.title('TIC-TAC-TOE---DataFlair')

digits = [1,2,3,4,5,6,7,8,9]

mark = '' “

count = 0

panels = ["panel"]\*10

* **Tk()** is use to initialize window
* **title()** used to set the title of the window

3. Function to check the result

def win(panels,sign):

**return** ((panels[1] == panels[2] == panels [3] == sign)

or (panels[1] == panels[4] == panels [7] == sign)

or (panels[1] == panels[5] == panels [9] == sign)

or (panels[2] == panels[5] == panels [8] == sign)

or (panels[3] == panels[6] == panels [9] == sign)

or (panels[3] == panels[5] == panels [7] == sign)

or (panels[4] == panels[5] == panels [6] == sign)

or (panels[7] == panels[8] == panels [9] == sign))

In this function, the result will be checked by checking which player makes three of their marks in a row (up, down, across, or diagonally).

4. Function to check the winner

def checker(digit):

global count, mark, digits

**if** digit==1 and digit **in** digits:

digits.remove(digit)

**if** count%2==0:

mark ='X'

panels[digit]=mark

elif count%2!=0:

mark = 'O'

panels[digit]=mar

button1.config(text = mark)

count = count+1

sign = mark

**if**(win(panels,sign) and sign=='X'):

msg.showinfo("Result","Player1 wins")

root.destroy()

elif(win(panels,sign) and sign=='O'):

msg.showinfo("Result","Player2 wins")

root.destroy()

**if** digit==2 and digit **in** digits:

digits.remove(digit)

**if** count%2==0:

mark ='X'

panels[digit]=mark

elif count%2!=0:

mark = 'O'

panels[digit]=mark

button2.config(text = mark)

count = count+1

sign = mark

**if**(win(panels,sign) and sign=='X'):

msg.showinfo("Result","Player1 wins")

root.destroy()

elif(win(panels,sign) and sign=='O'):

msg.showinfo("Result","Player2 wins")

root.destroy()

**if** digit==3 and digit **in** digits:

digits.remove(digit)

**if** count%2==0:

mark ='X'

panels[digit]=mark

elif count%2!=0:

mark = 'O'

panels[digit]=mark

button3.config(text = mark)

count = count+1

sign = mark

**if**(win(panels,sign) and sign=='X'):

msg.showinfo("Result","Player1 wins")

root.destroy()

elif(win(panels,sign) and sign=='O'):

msg.showinfo("Result","Player2 wins")

root.destroy()

**if** digit==4 and digit **in** digits:

digits.remove(digit)

**if** count%2==0:

mark ='X'

panels[digit]=mark

elif count%2!=0:

mark = 'O'

panels[digit]=mark

button4.config(text = mark)

count = count+1

sign = mark

**if**(win(panels,sign) and sign=='X'):

msg.showinfo("Result","Player1 wins")

root.destroy()

elif(win(panels,sign) and sign=='O'):

msg.showinfo("Result","Player2 wins")

root.destroy()

**if** digit==5 and digit **in** digits:

digits.remove(digit)

**if** count%2==0:

mark ='X'

panels[digit]=mark

elif count%2!=0:

mark = 'O'

panels[digit]=mark

button5.config(text = mark)

count = count+1

sign = mark

**if**(win(panels,sign) and sign=='X'):

msg.showinfo("Result","Player1 wins")

root.destroy()

elif(win(panels,sign) and sign=='O'):

msg.showinfo("Result","Player2 wins")

root.destroy()

**if** digit==6 and digit **in** digits:

digits.remove(digit)

**if** count%2==0:

mark ='X'

panels[digit]=mark

elif count%2!=0:

mark = 'O'

panels[digit]=mark

button6.config(text = mark)

count = count+1

sign

**if**(win(panels,sign) and sign=='X'):

msg.showinfo("Result","Player1 wins")

root.destroy()

elif(win(panels,sign) and sign=='O'):

msg.showinfo("Result","Player2 wins")

root.destroy()

**if** digit==7 and digit **in** digits:

digits.remove(digit)

**if** count%2==0:

mark ='X'

panels[digit]=mark

elif count%2!=0:

mark = 'O'

panels[digit]=mark

button7.config(text = mark)

count = count+1

sign = mark

**if**(win(panels,sign) and sign=='X'):

msg.showinfo("Result","Player1 wins")

root.destroy()

elif(win(panels,sign) and sign=='O'):

msg.showinfo("Result","Player2 wins")

root.destroy()

**if** digit==8 and digit **in** digits:

digits.remove(digit)

**if** count%2==0:

mark ='X'

panels[digit]=mark

elif count%2!=0:

mark = 'O'

panels[digit]=mark

button8.config(text = mark)

count = count+1

sign = mark

**if**(win(panels,sign) and sign=='X'):

msg.showinfo("Result","Player1 wins")

root.destroy()

elif(win(panels,sign) and sign=='O'):

msg.showinfo("Result","Player2 wins")

root.destroy()

**if** digit==9 and digit **in** digits:

digits.remove(digit)

**if** count%2==0:

mark ='X'

panels[digit]=mark

elif count%2!=0:

mark = 'O'

panels[digit]=mark

button9.config(text = mark)

count = count+1

sign = mark

**if**(win(panels,sign) and sign=='X'):

msg.showinfo("Result","Player1 wins")

root.destroy()

elif(win(panels,sign) and sign=='O'):

msg.showinfo("Result","Player2 wins")

root.destroy()

**if**(count>8 and win(panels,'X')==**False** and win(panels,'O')==**False**):

msg.showinfo("Result","Match Tied")

root.destroy()

Players have a total of 9 clicks to play the game. Each time the player clicked, one chance will decrease by increasing the value of count by 1 if the value of count is greater than 8 then the result of game is tie

* If the value of count is even then player1 will play else player2 will play.
* **config()** used to mark the button with appropriate text
* **showinfo()** methods in the messagebox widget used to show some relevant information
* **destroy()** stop the mainloop to quit the program

5. Define labels and buttons

Label(root,text="player1 : X",font="times 15").grid(row=0,column=1)

Label(root,text="player2 : O",font="times 15").grid(row=0,column=2)

button1=Button(root,width=15,font=('Times 16 bold'),height=7,command=lambda:checker(1))

button1.grid(row=1,column=1)

button2=Button(root,width=15,height=7,font=('Times 16 bold'),command=lambda:checker(2))

button2.grid(row=1,column=2)

button3=Button(root,width=15,height=7,font=('Times 16 bold'),command=lambda: checker(3))

button3.grid(row=1,column=3)

button4=Button(root,width=15,height=7,font=('Times 16 bold'),command=lambda: checker(4))

button4.grid(row=2,column=1)

button5=Button(root,width=15,height=7,font=('Times 16 bold'),command=lambda: checker(5))

button5.grid(row=2,column=2)

button6=Button(root,width=15,height=7,font=('Times 16 bold'),command=lambda: checker(6))

button6.grid(row=2,column=3)

button7=Button(root,width=15,height=7,font=('Times 16 bold'),command=lambda: checker(7))

button7.grid(row=3,column=1)

button8=Button(root,width=15,height=7,font=('Times 16 bold'),command=lambda: checker(8))

button8.grid(row=3,column=2)

button9=Button(root,width=15,height=7,font=('Times 16 bold'),command=lambda: checker(9))

button9.grid(row=3,column=3)

root.mainloop()

**Label()** widget used to display text that users aren’t able to modify.  
**Button()** widget display button

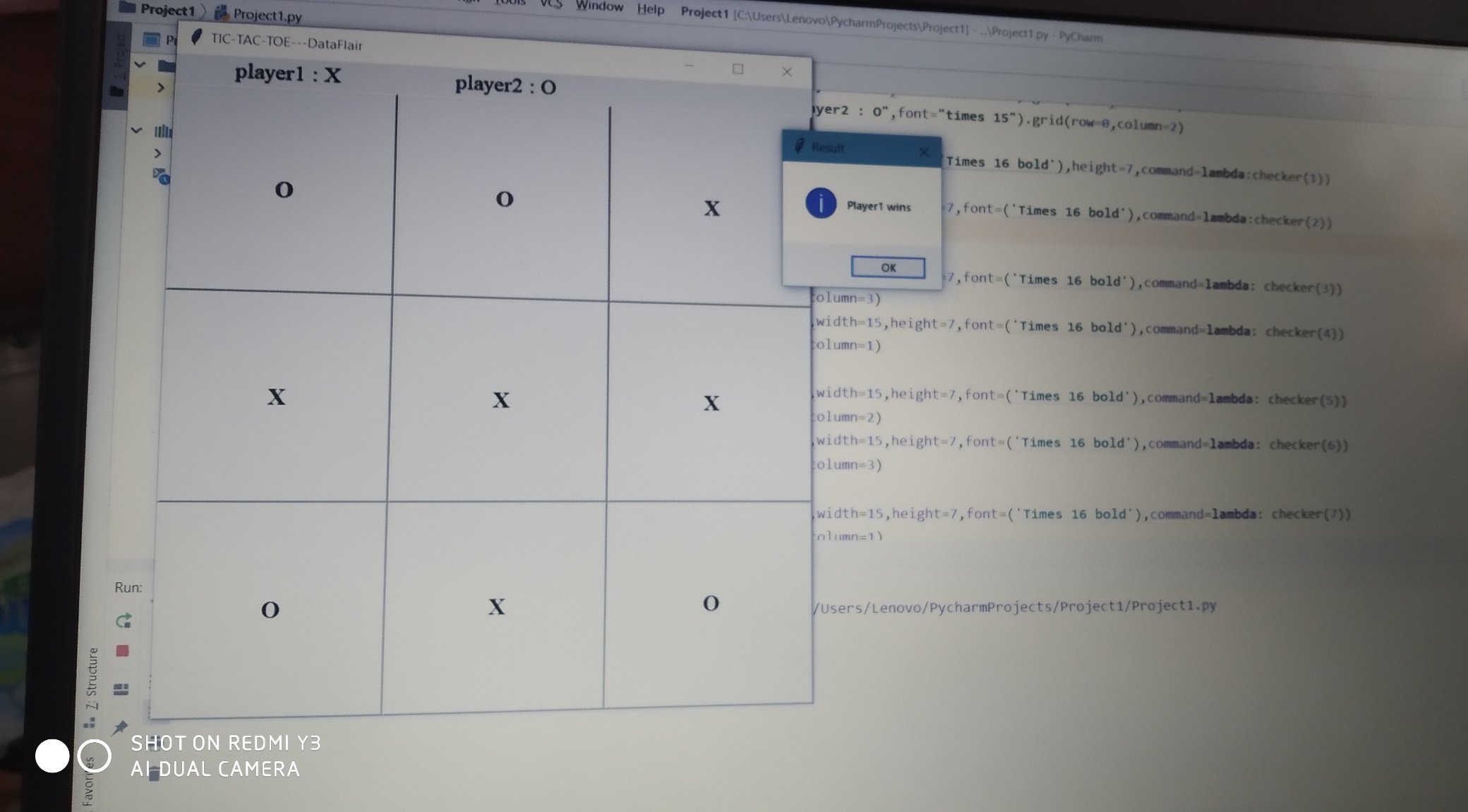
* **root** is the name of window which we refered
* **text** stores the value which we display on the label
* **font** in which our text is written
* **command** will called when the button is clicked
* **lambda()** function used to send specific data to the callback function.

**mainloop()** method executes when we want to run our program.

**Implementation of Program :**

**from** tkinter **import** \*  
**import** tkinter.messagebox **as** msg  
root= Tk()  
root.title(**'TIC-TAC-TOE---DataFlair'**)  
digits = [1,2,3,4,5,6,7,8,9]  
mark = **'' ''**count = 0  
panels = [**"panel"**]\*10  
  
**def** win(panels, sign):  
 **return** ((panels[1] == panels[2] == panels [3] == sign)  
 **or** (panels[1] == panels[4] == panels [7] == sign)  
 **or** (panels[1] == panels[5] == panels [9] == sign)  
 **or** (panels[2] == panels[5] == panels [8] == sign)  
 **or** (panels[3] == panels[6] == panels [9] == sign)  
 **or** (panels[3] == panels[5] == panels [7] == sign)  
 **or** (panels[4] == panels[5] == panels [6] == sign)  
 **or** (panels[7] == panels[8] == panels [9] == sign))  
  
  
**def** checker(digit):  
 **global** count, mark, digits  
 **if** digit == 1 **and** digit **in** digits:  
 digits.remove(digit)  
 **if** count % 2 == 0:  
 mark = **'X'** panels[digit] = mark  
 **elif** count % 2 != 0:  
 mark = **'O'** panels[digit] = mark  
 button1.config(text=mark)  
 count = count + 1  
 sign = mark  
 **if** (win(panels, sign) **and** sign == **'X'**):  
 msg.showinfo(**"Result"**, **"Player1 wins"**)  
 root.destroy()  
 **elif** (win(panels, sign) **and** sign == **'O'**):  
 msg.showinfo(**"Result"**, **"Player2 wins"**)  
 root.destroy()  
  
 **if** digit == 2 **and** digit **in** digits:  
 digits.remove(digit)  
 **if** count % 2 == 0:  
 mark = **'X'** panels[digit] = mark  
 **elif** count % 2 != 0:  
 mark = **'O'** panels[digit] = mark  
 button2.config(text=mark)  
 count = count + 1  
 sign = mark  
 **if** (win(panels, sign) **and** sign == **'X'**):  
 msg.showinfo(**"Result"**, **"Player1 wins"**)  
 root.destroy()  
 **elif** (win(panels, sign) **and** sign == **'O'**):  
 msg.showinfo(**"Result"**, **"Player2 wins"**)  
 root.destroy()  
  
 **if** digit == 3 **and** digit **in** digits:  
 digits.remove(digit)  
 **if** count % 2 == 0:  
 mark = **'X'** panels[digit] = mark  
 **elif** count % 2 != 0:  
 mark = **'O'** panels[digit] = mark  
 button3.config(text=mark)  
 count = count + 1  
 sign = mark  
 **if** (win(panels, sign) **and** sign == **'X'**):  
 msg.showinfo(**"Result"**, **"Player1 wins"**)  
 root.destroy()  
 **elif** (win(panels, sign) **and** sign == **'O'**):  
 msg.showinfo(**"Result"**, **"Player2 wins"**)  
 root.destroy()  
  
 **if** digit == 4 **and** digit **in** digits:  
 digits.remove(digit)  
 **if** count % 2 == 0:  
 mark = **'X'** panels[digit] = mark  
 **elif** count % 2 != 0:  
 mark = **'O'** panels[digit] = mark  
 button4.config(text=mark)  
 count = count + 1  
 sign = mark  
 **if** (win(panels, sign) **and** sign == **'X'**):  
 msg.showinfo(**"Result"**, **"Player1 wins"**)  
 root.destroy()  
 **elif** (win(panels, sign) **and** sign == **'O'**):  
 msg.showinfo(**"Result"**, **"Player2 wins"**)  
 root.destroy()  
  
 **if** digit == 5 **and** digit **in** digits:  
 digits.remove(digit)  
 **if** count % 2 == 0:  
 mark = **'X'** panels[digit] = mark  
 **elif** count % 2 != 0:  
 mark = **'O'** panels[digit] = mark  
 button5.config(text=mark)  
 count = count + 1  
 sign = mark  
 **if** (win(panels, sign) **and** sign == **'X'**):  
 msg.showinfo(**"Result"**, **"Player1 wins"**)  
 root.destroy()  
 **elif** (win(panels, sign) **and** sign == **'O'**):  
 msg.showinfo(**"Result"**, **"Player2 wins"**)  
 root.destroy()  
  
 **if** digit == 6 **and** digit **in** digits:  
 digits.remove(digit)  
 **if** count % 2 == 0:  
 mark = **'X'** panels[digit] = mark  
 **elif** count % 2 != 0:  
 mark = **'O'** panels[digit] = mark  
 button6.config(text=mark)  
 count = count + 1  
 sign=mark  
 **if** (win(panels, sign) **and** sign == **'X'**):  
 msg.showinfo(**"Result"**, **"Player1 wins"**)  
 root.destroy()  
 **elif** (win(panels, sign) **and** sign == **'O'**):  
 msg.showinfo(**"Result"**, **"Player2 wins"**)  
 root.destroy()  
  
  
  
 **if** digit == 7 **and** digit **in** digits:  
 digits.remove(digit)  
 **if** count % 2 == 0:  
 mark = **'X'** panels[digit] = mark  
 **elif** count % 2 != 0:  
 mark = **'O'** panels[digit] = mark  
 button7.config(text=mark)  
 count = count + 1  
 sign = mark  
 **if** (win(panels, sign) **and** sign == **'X'**):  
 msg.showinfo(**"Result"**, **"Player1 wins"**)  
 root.destroy()  
 **elif** (win(panels, sign) **and** sign == **'O'**):  
 msg.showinfo(**"Result"**, **"Player2 wins"**)  
 root.destroy()  
  
 **if** digit == 8 **and** digit **in** digits:  
 digits.remove(digit)  
 **if** count % 2 == 0:  
 mark = **'X'** panels[digit] = mark  
 **elif** count % 2 != 0:  
 mark = **'O'** panels[digit] = mark  
 button8.config(text=mark)  
 count = count + 1  
 sign = mark  
 **if** (win(panels, sign) **and** sign == **'X'**):  
 msg.showinfo(**"Result"**, **"Player1 wins"**)  
 root.destroy()  
 **elif** (win(panels, sign) **and** sign == **'O'**):  
 msg.showinfo(**"Result"**, **"Player2 wins"**)  
 root.destroy()  
  
 **if** digit == 9 **and** digit **in** digits:  
 digits.remove(digit)  
 **if** count % 2 == 0:  
 mark = **'X'** panels[digit] = mark  
 **elif** count % 2 != 0:  
 mark = **'O'** panels[digit] = mark  
 button9.config(text=mark)  
 count = count + 1  
 sign = mark  
 **if** (win(panels, sign) **and** sign == **'X'**):  
 msg.showinfo(**"Result"**, **"Player1 wins"**)  
 root.destroy()  
 **elif** (win(panels, sign) **and** sign == **'O'**):  
 msg.showinfo(**"Result"**, **"Player2 wins"**)  
 root.destroy()  
  
 **if** (count > 8 **and** win(panels, **'X'**) == **False and** win(panels, **'O'**) == **False**):  
 msg.showinfo(**"Result"**, **"Match Tied"**)  
 root.destroy()  
  
  
Label(root,text=**"player1 : X"**,font=**"times 15"**).grid(row=0,column=1)  
Label(root,text=**"player2 : O"**,font=**"times 15"**).grid(row=0,column=2)  
  
button1=Button(root,width=15,font=(**'Times 16 bold'**),height=7,command=**lambda**:checker(1))  
button1.grid(row=1,column=1)  
button2=Button(root,width=15,height=7,font=(**'Times 16 bold'**),command=**lambda**:checker(2))  
button2.grid(row=1,column=2)  
  
button3=Button(root,width=15,height=7,font=(**'Times 16 bold'**),command=**lambda**: checker(3))  
button3.grid(row=1,column=3)  
button4=Button(root,width=15,height=7,font=(**'Times 16 bold'**),command=**lambda**: checker(4))  
button4.grid(row=2,column=1)  
  
button5=Button(root,width=15,height=7,font=(**'Times 16 bold'**),command=**lambda**: checker(5))  
button5.grid(row=2,column=2)  
button6=Button(root,width=15,height=7,font=(**'Times 16 bold'**),command=**lambda**: checker(6))  
button6.grid(row=2,column=3)  
  
button7=Button(root,width=15,height=7,font=(**'Times 16 bold'**),command=**lambda**: checker(7))  
button7.grid(row=3,column=1)  
button8=Button(root,width=15,height=7,font=(**'Times 16 bold'**),command=**lambda**: checker(8))  
button8.grid(row=3,column=2)  
  
button9=Button(root,width=15,height=7,font=(**'Times 16 bold'**),command=**lambda**: checker(9))  
button9.grid(row=3,column=3)  
  
  
root.mainloop()

**Output :-**

****

**Conclusion :**

Hence in this Mini project we learn how to create buttons and config text on buttons and also how to use lambda functions to send specific values to callback functions. In this way we successfully made a Tic-Tac-Toe game python project.