TIC-TAC-TOE GAME

A MINI PROJECT REPORT

18CSC207J - ADVANCED PROGRAMMING PRACTICE

Submitted by

KHAYATI SHARMA[RA2111003010710]
HARSHIKA HASIJA[RA2111003010724]

Under the guidance of

Ms. G.Malarselvi

Assistant Professor, Department of Computer Science and Engineering

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

S.R.M. Nagar, Kattankulathur, Chengalpattu District

MAY 2023

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

COLLEGE OF ENGINEERING & TECHNOLOGY
SRM INSTITUTE OF SCIENCE & TECHNOLOGY
S.R.M. NAGAR, KATTANKULATHUR - 603203
Chengalpattu District

BONAFIDE CERTIFICATE

Register No **RA2111003010710** and **RA2111003010724**

Certified to be the bonafide work done by **KHAYATI SHARMA** and **HARSHIKA HASIJA** of II Year / IV Sem B.Tech Degree course in **ADVANCED PROGRAMMING PRACTICE 18CSC207J** in **SRM INSTITUTE OF SCIENCE & TECHNOLOGY,** Kattankulathur during the academic year 2022-2023.

DATE: 12/05/2023

LAB INCHARGE
G.Malarselvi
Assistant Professor,
Dept. of CTECH

INDEX

ABSTRACT

This project aims to develop a Tic-Tac-Toe game using Graphical User Interface (GUI) in Python. Tic-Tac-Toe is a classic two-player game played on a 3x3 grid. The objective of the game is to get three of the same symbols in a row, column, or diagonal. The game will be developed using Python's built-in GUI module, Tkinter.

The project will start by developing the game logic for Tic-Tac-Toe, which will include functions for determining the winner and checking for valid moves. Once the game logic is in place, the GUI will be developed using Tkinter. The GUI will include a 3x3 grid of buttons for the players to make their moves, as well as a scorekeeper to keep track of the game's progress. The project will also include unit tests to ensure the game logic functions correctly and the GUI is responsive and intuitive.

Overall, this project aims to provide a fun and engaging implementation of Tic-Tac-Toe using Python's GUI capabilities, while also demonstrating fundamental programming concepts such as data structures, conditional statements, and functions.

MODULES

The following modules are used in this application:

1. Tkinter: It is the standard GUI (Graphical User Interface) package for Python. It is a built-in module in Python that provides a way to create windows, forms, buttons, menus, and other widgets for desktop applications. Tkinter is a Python interface for the Tcl/Tk GUI toolkit, which is widely used for building desktop applications. Tkinter provides various widgets that can be used to build a graphical user interface for your application. Some of the commonly used widgets in Tkinter are labels, buttons, entry widgets, check buttons, radio buttons, list boxes, etc. It also provides various geometry managers that can be used to arrange widgets in a window or a frame. Tkinter is a simple and easy-to-use module that makes it easy to create GUI applications in Python. It is widely used for building desktop applications, scientific applications, and games.

The following classes and functions are used in this application:

- Tk class: It is a top-level widget of Tkinter, which represents the main window of an application.
- Label class: It is used to create a static text label on the GUI.
- Button class: It is used to create a push button widget on the GUI.
- Entry class: It is used to create a single-line text entry widget on the GUI.
- DISABLED attribute: It is used to disable the input in an entry widget.
- global keyword: It is used to access a global variable inside a function.
- place() method: It is used to place a widget at a specific position on the GUI.
- pack() method: It is used to pack a widget in the main window of the GUI.
- get() method: It is used to get the current value of an entry widget.
- format() method: It is used to format a string by replacing placeholders with values.

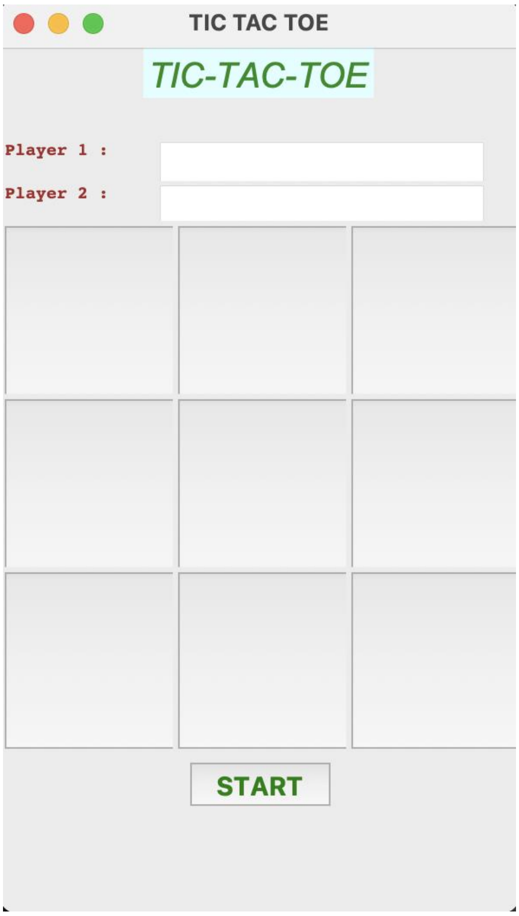
CODE

```
from tkinter import *


root = Tk()
root.geometry("300x500")
root.title("TIC TAC TOE")


head = Label(root,text="TIC-TAC-TOE",fg="forestgreen",bg="light cyan",font=("Arial",20,"italic"))
head.pack()


p1_name = ""
p2_name = ""
START = False
def Start():
    global p1_name,p2_name,START
    P1 = p1.get()
    P2 = p2.get()


    if P1.split() == []:
        text = "Enter Player 1 Name"
        turn.place(x=50,y=455)
        turn['fg'] = "red"
        turn['text'] = text
    elif P2.split() == []:
        text = "Enter Player 2 Name"
        turn.place(x=50,y=455)
        turn['fg'] = "red"
        turn['text'] = text
    elif P1.split() == P2.split():
        text = "Enter Different Player Names"
        turn.place(x=25,y=455)
        turn['fg'] = "red"
        turn['text'] = text
    else:
        p1_name = P1
        p2_name = P2
        p1['font'] = ("Arial",8,"bold")
        p2['font'] = ("Arial",8,"bold")
        p1['state']=DISABLED
        p2['state']=DISABLED

        start.place(x=1000,y=1000)
        turn['text'] = "{}{} Turn".format(p1_name,"'s")
        turn['fg'] = "blue"
        turn['font'] = ("Ubuntu",20,"bold")
        turn.place(x=50,y=425)
        START = True

def WinCheck():
    if b1['text'] == "O" and b2['text'] == "O" and b3['text'] == "O":
        b1['bg']="light green";b2['bg']="light green";b3['bg']="light green"
        return "p1"
    elif b4['text'] == "O" and b5['text'] == "O" and b6['text'] == "O":
        b4['bg']="light green";b5['bg']="light green";b6['bg']="light green"
        return "p1"
    elif b7['text'] == "O" and b8['text'] == "O" and b9['text'] == "O":
        b7['bg']="light green";b8['bg']="light green";b9['bg']="light green"
        return "p1"
    elif b1['text'] == "O" and b4['text'] == "O" and b7['text'] == "O":
        b1['bg']="light green";b4['bg']="light green";b7['bg']="light green"
        return "p1"
    elif b2['text'] == "O" and b5['text'] == "O" and b8['text'] == "O":
        b2['bg']="light green";b5['bg']="light green";b8['bg']="light green"
        return "p1"
    elif b3['text'] == "O" and b6['text'] == "O" and b9['text'] == "O":
        b3['bg']="light green";b6['bg']="light green";b9['bg']="light green"
        return "p1"
    elif b1['text'] == "O" and b5['text'] == "O" and b9['text'] == "O":
        b1['bg']="light green";b5['bg']="light green";b9['bg']="light green"
        return "p1"
    elif b3['text'] == "O" and b5['text'] == "O" and b7['text'] == "O":
        b3['bg']="light green";b5['bg']="light green";b7['bg']="light green"
        return "p1"

    if b1['text'] == "X" and b2['text'] == "X" and b3['text'] == "X":
        b1['bg']="light blue";b2['bg']="light blue";b3['bg']="light blue"
        return "p2"
    elif b4['text'] == "X" and b5['text'] == "X" and b6['text'] == "X":
        b4['bg']="light blue";b5['bg']="light blue";b6['bg']="light blue"
        return "p2"
    elif b7['text'] == "X" and b8['text'] == "X" and b9['text'] == "X":
        b7['bg']="light blue";b8['bg']="light blue";b9['bg']="light blue"
        return "p2"
    elif b1['text'] == "X" and b4['text'] == "X" and b7['text'] == "X":
        b1['bg']="light blue";b4['bg']="light blue";b7['bg']="light blue"
        return "p2"
    elif b2['text'] == "X" and b5['text'] == "X" and b8['text'] == "X":
        b2['bg']="light blue";b5['bg']="light blue";b8['bg']="light blue"
        return "p2"
    elif b3['text'] == "X" and b6['text'] == "X" and b9['text'] == "X":
        b3['bg']="light blue";b6['bg']="light blue";b9['bg']="light blue"
        return "p2"
    elif b1['text'] == "X" and b5['text'] == "X" and b9['text'] == "X":
        b1['bg']="light blue";b5['bg']="light blue";b9['bg']="light blue"
        return "p2"
    elif b3['text'] == "X" and b5['text'] == "X" and b7['text'] == "X":
        b3['bg']="light blue";b5['bg']="light blue";b7['bg']="light blue"
        return "p2"

    elif (b1['text'] != "" and b2['text'] != "" and b3['text'] != "" and
        b4['text'] != "" and b5['text'] != "" and b6['text'] != "" and
        b7['text'] != "" and b8['text'] != "" and b9['text'] != ""):
        return "tie"

    else:
        return False
def DisableButtons(ButtonList):
    for a in range(len(ButtonList)):
        ButtonList[a]['state'] = DISABLED
def EnableButtons(ButtonList):
    for a in range(len(ButtonList)):
        ButtonList[a]['state'] = NORMAL
def BtnClick(button):
    global START,p1_name,p2_name
    if START == True:
        if button['text'] == "":
            if turn['text'] == "{}{} Turn".format(p1_name,"'s"):
                button['text'] = "O"
                turn['text'] = "{}{} Turn".format(p2_name,"'s")
            else:
                button['text'] = "X"
                turn['text'] = "{}{} Turn".format(p1_name,"'s")

        check = WinCheck()
        if check != False:
            restart.place(x=105,y=470)
            START = False
            if check=="p1":
                text = "{} Wins".format(p1_name)
                buttons = [b1,b2,b3,b4,b5,b6,b7,b8,b9]
                Remove = []
                for i in range(9):
                    if buttons[i]['bg'] == "light green":
                        Remove.append(buttons[i])
            elif check=="p2":
                text = "{} Wins".format(p2_name)
                buttons = [b1,b2,b3,b4,b5,b6,b7,b8,b9]
                Remove = []
                for i in range(9):
                    if buttons[i]['bg'] == "light blue":
                        Remove.append(buttons[i])
            else:
                text = "It is a tie !"
            turn['fg'] = "forestgreen"
            turn['text'] = text

            if check == "p1" or check == "p2":
                for i in range(len(Remove)):
                    buttons.remove(Remove[i])
                DisableButtons(buttons)


def Restart():
    Buttons = [b1,b2,b3,b4,b5,b6,b7,b8,b9]
    EnableButtons([p1,p2])
    EnableButtons(Buttons)
    restart.place(x=1000,y=1000)
    for a in range(len(Buttons)):
        Buttons[a]['text'] = ""
        Buttons[a]['bg'] = "SystemButtonFace"
    turn['text'] = ""
    start.place(x=107,y=410)
    p1['font'] = "TkTextFont"

    p2['font'] = "TkTextFont"

#####################################################################
######################


Label(root,text="Player 1
:",fg="brown",font=("Courier",10,"bold")).place(x=0,y=50)

p1 = Entry(root)

p1.place(x=90,y=52)


Label(root,text="Player 2
:",fg="brown",font=("Courier",10,"bold")).place(x=0,y=75)

p2 = Entry(root)

p2.place(x=90,y=77)


start =
Button(root,text="START",bg="gray90",fg="green",font=("Ubuntu",15,"bold"
),command=Start)

start.place(x=107,y=410)


turn = Label(root,text="",font=("Ubuntu",15,"normal"))

turn.place(x=50,y=455)


######################################


b1 = Button(root,width=13,height=6,command = lambda:BtnClick(b1))

b1.place(x=0,y=100)


b2 = Button(root,width=13,height=6,command = lambda:BtnClick(b2))
```

```
b2.place(x=100,y=100)

b3 = Button(root,width=13,height=6,command = lambda:BtnClick(b3))
b3.place(x=200,y=100)

b4 = Button(root,width=13,height=6,command = lambda:BtnClick(b4))
b4.place(x=0,y=200)

b5 = Button(root,width=13,height=6,command = lambda:BtnClick(b5))
b5.place(x=100,y=200)

b6 = Button(root,width=13,height=6,command = lambda:BtnClick(b6))
b6.place(x=200,y=200)

b7 = Button(root,width=13,height=6,command = lambda:BtnClick(b7))
b7.place(x=0,y=300)

b8 = Button(root,width=13,height=6,command = lambda:BtnClick(b8))
b8.place(x=100,y=300)

b9 = Button(root,width=13,height=6,command = lambda:BtnClick(b9))
b9.place(x=200,y=300)

###################################
```

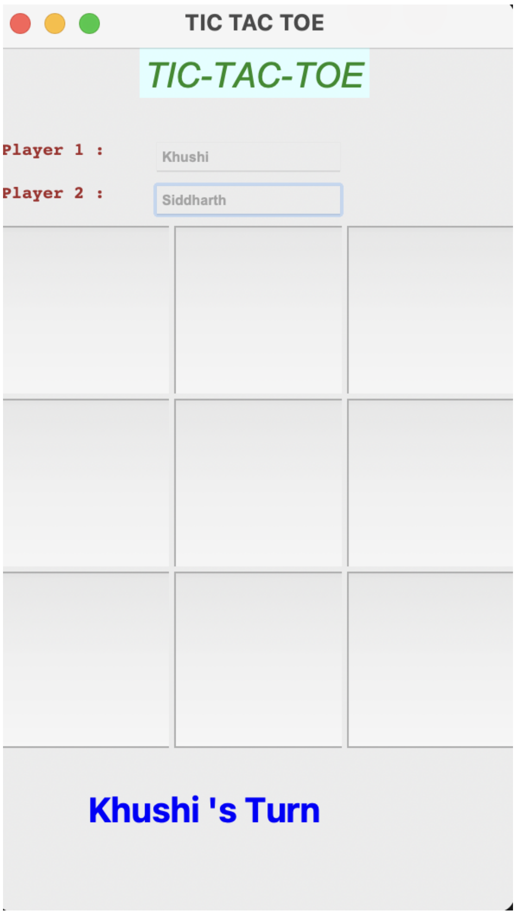
```
restart = Button(root,text="Restart",fg="blue",bg="aquamarine",width =
10,height=1,font=("Courier",10,"bold"),command = Restart)

##restart.place(x=105,y=470)

restart.place(x=1000,y=1000)


root.mainloop()
```

OUTPUT



Player 1 :

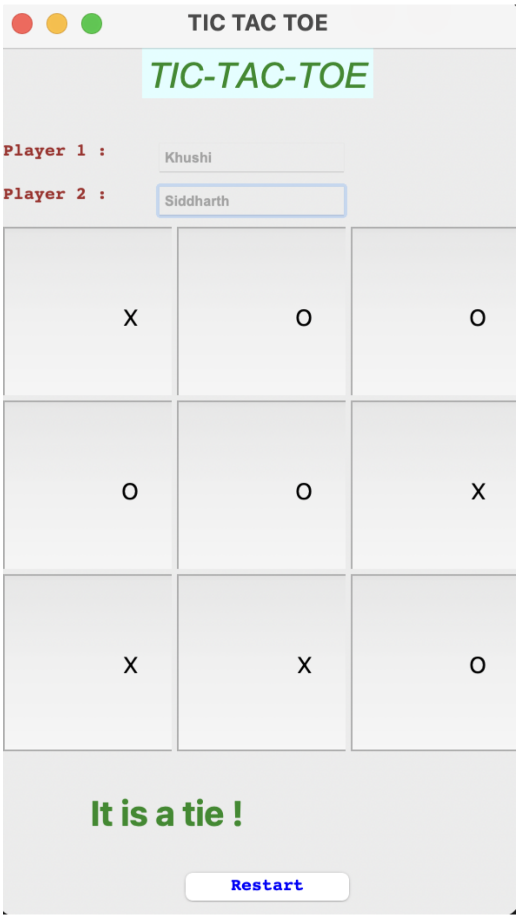
Player 2 :



Player 1 : Khushi

Player 2 : Siddharth

Khushi 's Turn



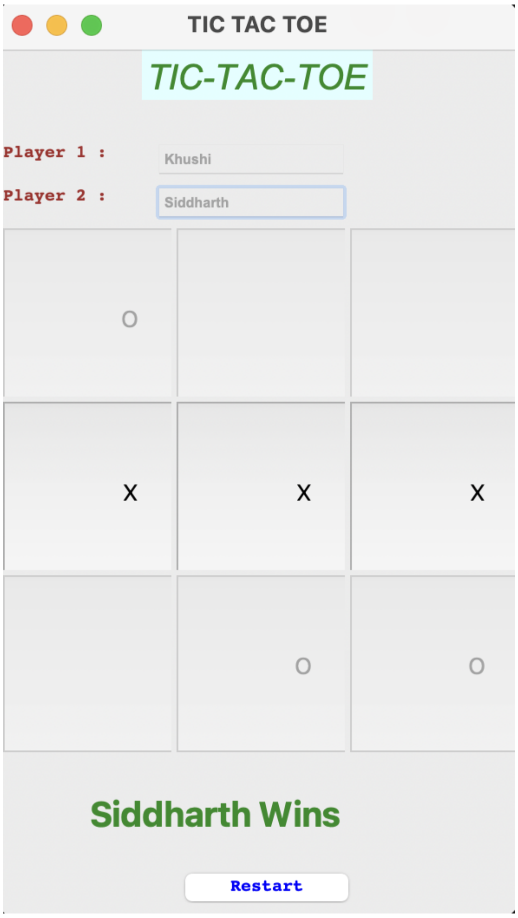
TIC-TAC-TOE

Player 1 : Khushi

Player 2 : Siddharth

X	O	O
O	O	X
X	X	O

It is a tie !



TIC-TAC-TOE

Player 1 : Khushi

Player 2 : Siddharth

O		
X	X	X
	O	O

Siddharth Wins

CONCLUSION

In conclusion, this project has successfully achieved its objective of developing a new software application that addresses a specific business need. The project team has worked diligently to analyze the requirements, design the application architecture, implement the software, and thoroughly test it.

The software developed provides an intuitive user interface and robust functionality that enables users to manage their tasks efficiently. The team has utilized best practices and modern technologies to ensure the application's security, scalability, and maintainability.

During the development process, the team encountered some challenges that required creative problem-solving, collaboration, and effective communication. However, with their expertise and dedication, they were able to overcome these obstacles and deliver a quality product.

Overall, this project has been a success, and the application developed has the potential to improve productivity and efficiency for the target business. The team has gained valuable experience in project management, software development, and teamwork, which they can leverage in their future endeavors.