

COM301T&P: OS PROJECT

TIC-TAC-TOE

TEAM MEMBERS:

COE18B025	- JAGADHABHIRAM PRATIKANTAM
COE18B026	- JAYANTH S
COE18B039	- VISHNU SREE SIRI
COE18B050	- SONA ELZA SIMON
CED18I053	- THOTA SAI KEERTHANA

OVERVIEW:

In this project, we made a tic-tac-toe game where a user can play a game with a computer or along with another user on the system. We used python programming language and tkinter for GUI.

About the game:

It is a game for two players, X and O, who take turns marking the spaces in a 3×3 grid. Players take turns putting their marks in empty squares. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row is the winner.

Modes of playing:

1. Player v/s Player
2. Player v/s Computer

Possible end scenarios:

1. Win or lose
2. Draw

Additional features:

1. Statistics: Displays the winners details on the terminal
2. Each move made by the user is also displayed at the terminal.



Code:

```
from tkinter import *
from tkinter import messagebox
import random
import sys
import threading

root = Tk()
name3=None
name4=None
turn=1
flag=1
root.title("WELCOME")
root.geometry("300x300")

board=[i for i in range(0,9)]
player_copy, computer_copy = 'X','O'

def make_move(brd, player, move, undo=False):
    if can_move(brd, player, move):
        brd[move-1] = player
        win=can_win(brd, player, move)
        if undo:
            brd[move-1] = move-1
        return (True, win)
    return (False, False)

def can_move(brd, player, move):
    tab=range(1,10)
    if move in tab and brd[move-1] == move-1:
        return True
    return False

def can_win(brd, player, move):
    win=True
    winners=((0,1,2),(3,4,5),(6,7,8),(0,3,6),(1,4,7),(2,5,8),(0,4,8),(2,4,6))
    for tup in winners:
        win=True
        for ix in tup:
            if brd[ix] != player:
                win=False
```

```

        break
    if win == True:
        break
    return win

def maximo():
    global board
    global player_copy
    global computer_copy
    moves=((5,),(1,7,3,9),(2,4,6,8))
    move=-1
    # If I can win, others don't matter.
    for i in range(1,10):
        if make_move(board, computer_copy, i, True)[1]:
            move=i
            break
    if move == -1:
        # If player can win, block him.
        for i in range(1,10):
            if make_move(board, player_copy, i, True)[1]:
                move=i
                break
    if move == -1:
        # Otherwise, try to take one of desired places.
        for tup in moves:
            for mv in tup:
                if move == -1 and can_move(board, computer_copy, mv):
                    move=mv
                    break
    make_move(board, computer_copy, move)
    return move

def open_stat():
    f=open("stat.txt","r")
    if f.mode=="r":
        contents=f.read()
        print(contents)

def info2():
    global name3
    def show_entry_fields():
        global name3

```

```

        name3 = name1.get()
        w1.destroy()
        CP_win()
w1 = Tk()
w1.title("ENTER YOUR DETAILS ")
w1.geometry("400x150")
w1.configure(bg="lightgreen")
lbl=Label(w1,text= "Player1:
",bg='lightgreen',font=('Roman','10','bold')).place(relx=0.05,rely=0.2)
name1 = Entry(w1)

name1.place(relx=0.3,rely=0.2)
btn=Button(w1,
            text='Enter',font=('Roman','10','bold'),bg='black',fg='yellow',
            command=show_entry_fields)
btn.place(relx=0.5,rely=0.7,anchor=CENTER)

w1.mainloop()

def CP_win():
    global name3
    global board
    window = Toplevel(root)
    window.title("TIC-TAC-TOE")
    window.geometry("600x500")
    window.configure(bg="lightblue")
    lbl=Label(window,text="Tic-tac-toe
Game",font=('Times','20','bold'),bg='lightblue')
    lbl.grid(row=0,column=0)
    lbl=Label(window,text=name3+"
X",font=('Times','15','bold'),fg='darkgreen',bg="lightblue")
    lbl.grid(row=5,column=0)
    lbl=Label(window,text="Computer"+"
O",font=('Times','15','bold'),fg='darkgreen',bg="lightblue")
    lbl.grid(row=6,column=0)
    print("Player vs Computer")
    def clicked1():
        global turn
        if btn1["text"]==" ":
            if turn==1:
                turn =2
                btn1["text"]="X"

```

```

        board[0] = 'X'
        print("Player played ",btn1["text"]," in grid 1")
    elif turn==2:
        turn=1
        btn1["text"]="O"
    check()

def clicked2():
    global turn
    if btn2["text"]==" ":
        if turn==1:
            turn =2
            btn2["text"]="X"
            board[1] = 'X'
            print("Player played ",btn2["text"]," in grid 2")
        elif turn==2:
            turn=1
            btn2["text"]="O"
    check()

def clicked3():
    global turn
    if btn3["text"]==" ":
        if turn==1:
            turn =2
            btn3["text"]="X"
            board[2] = 'X'
            print("Player played ",btn3["text"]," in grid 3")
        elif turn==2:
            turn=1
            btn3["text"]="O"
    check()

def clicked4():
    global turn
    if btn4["text"]==" ":
        if turn==1:
            turn =2
            btn4["text"]="X"
            board[3] = 'X'
            print("Player played ",btn4["text"]," in grid 4")
        elif turn==2:

```

```

        turn=1
        btn4["text"]="O"
        check()

def clicked5():
    global turn
    if btn5["text"]==" ":
        if turn==1:
            turn =2
            btn5["text"]="X"
            board[4] = 'X'
            print("Player played ",btn5["text"]," in grid 5")
        elif turn==2:
            turn=1
            btn5["text"]="O"
        check()

def clicked6():
    global turn
    if btn6["text"]==" ":
        if turn==1:
            turn =2
            btn6["text"]="X"
            board[5] = 'X'
            print("Player played ",btn6["text"]," in grid 6")
        elif turn==2:
            turn=1
            btn6["text"]="O"
        check()

def clicked7():
    global turn
    if btn7["text"]==" ":
        if turn==1:
            turn =2
            btn7["text"]="X"
            board[6] = 'X'
            print("Player played ",btn7["text"]," in grid 7")
        elif turn==2:
            turn=1
            btn7["text"]="O"
        check()

```

```

def clicked8():
    global turn
    if btn8["text"]==" ":
        if turn==1:
            turn =2
            btn8["text"]="X"
            board[7] = 'X'
            print("Player played ",btn8["text"]," in grid 8")
        elif turn==2:
            turn=1
            btn8["text"]="O"
        check()

def clicked9():
    global turn
    if btn9["text"]==" ":
        if turn==1:
            turn =2
            btn9["text"]="X"
            board[8] = 'X'
            print("Player played ",btn9["text"]," in grid 9")
        elif turn==2:
            turn=1
            btn9["text"]="O"
        check()

def check():#checking for the winner
    global flag
    global turn
    b1 = btn1["text"]
    b2 = btn2["text"]
    b3 = btn3["text"]
    b4 = btn4["text"]
    b5 = btn5["text"]
    b6 = btn6["text"]
    b7 = btn7["text"]
    b8 = btn8["text"]
    b9 = btn9["text"]
    flag=flag+1

def min1():

```



```

        if b1==b2 and b1==b3 and b1=="O" or b1==b2 and b1==b3 and b1=="X":
            win(btn1["text"])
def min2():
    if b4==b5 and b4==b6 and b4=="O" or b4==b5 and b4==b6 and b4=="X":
        win(btn4["text"])
def min3():
    if b7==b8 and b7==b9 and b7=="O" or b7==b8 and b7==b9 and b7=="X":
        win(btn7["text"])
def min4():
    if b1==b4 and b1==b7 and b1=="O" or b1==b4 and b1==b7 and b1=="X":
        win(btn1["text"])
def min5():
    if b2==b5 and b2==b8 and b2=="O" or b2==b5 and b2==b8 and b2=="X":
        win(btn2["text"])
def min6():
    if b3==b6 and b3==b9 and b3=="O" or b3==b6 and b3==b9 and b3=="X":
        win(btn3["text"])
def min7():
    if b1==b5 and b1==b9 and b1=="O" or b1==b5 and b1==b9 and b1=="X":
        win(btn1["text"])
def min8():
    if b7==b5 and b7==b3 and b7=="O" or b7==b5 and b7==b3 and b7=="X":
        win(btn7["text"])

```

#multithreading - 8 threads - to check for all patterns(8) of winning the game

```

t1 = threading.Thread(target=min1)
t2 = threading.Thread(target=min2)
t3 = threading.Thread(target=min3)
t4 = threading.Thread(target=min4)
t5 = threading.Thread(target=min5)
t6 = threading.Thread(target=min6)
t7 = threading.Thread(target=min7)
t8 = threading.Thread(target=min8)
t1.start()
t2.start()
t3.start()
t4.start()
t5.start()
t6.start()
t7.start()
t8.start()

```

```

#when the all the 9 boxes are filled with value and no one wins - tied
if flag ==10:
    messagebox.showinfo("Tie", "Match Tied!!! Try again :)")
    #setting values back for new game
    turn=1
    flag=1
    global board
    board = [i for i in range(0,9)]
    print("The game is over !")
    print("Match tied")
    window.destroy()

#computer's turn to play
if(turn == 2):
    x = maximo()
    print("Computer's turn: Plays grid ", x)
    if(x == 1):
        clicked1()
    elif(x == 2):
        clicked2()
    elif(x == 3):
        clicked3()
    elif(x == 4):
        clicked4()
    elif(x == 5):
        clicked5()
    elif(x == 6):
        clicked6()
    elif(x == 7):
        clicked7()
    elif(x == 8):
        clicked8()
    elif(x == 9):
        clicked9()

#saving the winner details
def win(player):
    global turn
    global flag
    file = open("stat.txt","a")
    print("The game is over !")

```

```

    if player=="X":
        ans = "Game complete " +name3 + " wins "
        winner= "PVC: "+name3+" wins\n"
        print(name3," wins")
    else :
        ans = "Game complete Computer wins "
        winner= "PVC: Computer wins\n"
        print("Computer wins")

    file.write(winner)
    messagebox.showinfo("Congratulations", ans)
    #setting values back to default for new game
    turn=1
    flag=1
    global board
    board = [i for i in range(0,9)]
    window.destroy() # close the program


    btn1 = Button(window, text=" ",bg="black",
fg="white",width=3,height=2,font=('Roman','20'),command=clicked1)
    btn1.grid(column=1, row=7)
    btn2 = Button(window, text=" ",bg="black",
fg="white",width=3,height=2,font=('Roman','20'),command=clicked2)
    btn2.grid(column=2, row=7)
    btn3 = Button(window, text=" ",bg="black",
fg="white",width=3,height=2,font=('Roman','20'),command=clicked3)
    btn3.grid(column=3, row=7)
    btn4 = Button(window, text=" ",bg="black",
fg="white",width=3,height=2,font=('Roman','20'),command=clicked4)
    btn4.grid(column=1, row=8)
    btn5 = Button(window, text=" ",bg="black",
fg="white",width=3,height=2,font=('Roman','20'),command=clicked5)
    btn5.grid(column=2, row=8)
    btn6 = Button(window, text=" ",bg="black",
fg="white",width=3,height=2,font=('Roman','20'),command=clicked6)
    btn6.grid(column=3, row=8)
    btn7 = Button(window, text=" ",bg="black",
fg="white",width=3,height=2,font=('Roman','20'),command=clicked7)
    btn7.grid(column=1, row=9)
    btn8 = Button(window, text=" ",bg="black",
fg="white",width=3,height=2,font=('Roman','20'),command=clicked8)
    btn8.grid(column=2, row=9)

```

```

    btn9 = Button(window, text=" ",bg="black",
fg="white",width=3,height=2,font=('Roman','20'),command=clicked9)
    btn9.grid(column=3, row=9)

    window.mainloop()

def info1():
    global name3
    global name4
    def show_entry_fields():
        global name3
        global name4
        name3=name1.get()
        name4=name2.get()
        w1.destroy()
        twoP_win()

    w1 = Tk()

    w1.title("ENTER YOUR DETAILS ")
    w1.geometry("400x150")
    w1.configure(bg='lightgreen')
    lbl=Label(w1,bg='lightgreen',text= "Player1:
",font=('Roman','10','bold')).place(relx=0,rely=0.1)
    lbl=Label(w1,bg='lightgreen',text= "Player2:
",font=('Roman','10','bold')).place(relx=0,rely=0.3)
    name1 = Entry(w1)
    name2 = Entry(w1)

    name1.place(relx=0.25,rely=0.1)
    name2.place(relx=0.25,rely=0.3)
    btn=Button(w1, text='Enter',
font=('Roman','10','bold'),bg='black',fg='yellow',command=show_entry_fields)
    btn.place(relx=0.5,rely=0.73,anchor=CENTER)

    w1.mainloop()

def twoP_win(): # two player window definition
    global name3
    global name4
    window = Toplevel(root)
    window.title("TIC-TAC-TOE")

```

```

window.geometry("600x500")
window.configure(bg="lightblue")
lbl=Label(window,text="Tic-tac-toe
Game",font=('Times','20','bold'),bg='lightblue')
lbl.grid(row=0,column=0)
lbl=Label(window,text=name3+" :
X",font=('Times','15','bold'),bg='lightblue',fg='darkgreen')
lbl.grid(row=5,column=0)
lbl=Label(window,text=name4+" :
O",font=('Times','15','bold'),bg='lightblue',fg='darkgreen')
lbl.grid(row=6,column=0)

print("Player vs Player")

played=0
def clicked1():
    global turn
    global played
    played=1
    if btn1["text"]==" ":
        if turn==1:
            turn =2
            btn1["text"]="X"
        elif turn==2:
            turn=1
            btn1["text"]="O"
        print("Player played ",btn1["text"]," in grid 1")
        check()

def clicked2():
    global played
    global turn
    played=2
    if btn2["text"]==" ":
        if turn==1:
            turn =2
            btn2["text"]="X"
        elif turn==2:
            turn=1
            btn2["text"]="O"
        print("Player played ",btn2["text"]," in grid 2")
        check()

```

```

def clicked3():
    global played
    global turn
    played=3
    if btn3["text"]==" ":
        if turn==1:
            turn =2
            btn3["text"]="X"
        elif turn==2:
            turn=1
            btn3["text"]="O"
    print("Player played ",btn3["text"]," in grid 3")
    check()

def clicked4():
    global played
    global turn
    played=4
    if btn4["text"]==" ":
        if turn==1:
            turn =2
            btn4["text"]="X"
        elif turn==2:
            turn=1
            btn4["text"]="O"
    print("Player played ",btn4["text"]," in grid 4")
    check()

def clicked5():
    global played
    global turn
    played=5
    if btn5["text"]==" ":
        if turn==1:
            turn =2
            btn5["text"]="X"
        elif turn==2:
            turn=1
            btn5["text"]="O"
    print("Player played ",btn5["text"]," in grid 5")
    check()

```

```

def clicked6():
    global played
    global turn
    played=6
    if btn6["text"]==" ":
        if turn==1:
            turn =2
            btn6["text"]="X"
        elif turn==2:
            turn=1
            btn6["text"]="O"
    print("Player played ",btn6["text"]," in grid 6")
    check()

def clicked7():
    global played
    global turn
    played=7
    if btn7["text"]==" ":
        if turn==1:
            turn =2
            btn7["text"]="X"
        elif turn==2:
            turn=1
            btn7["text"]="O"
    print("Player played ",btn7["text"]," in grid 7")
    check()

def clicked8():
    global played
    global turn
    played=8
    if btn8["text"]==" ":
        if turn==1:
            turn =2
            btn8["text"]="X"
        elif turn==2:
            turn=1
            btn8["text"]="O"
    print("Player played ",btn8["text"]," in grid 8")
    check()

```

```

def clicked9():
    global played
    global turn
    played=9
    if btn9["text"]==" ":
        if turn==1:
            turn =2
            btn9["text"]="X"
        elif turn==2:
            turn=1
            btn9["text"]="O"
        print("Player played ",btn9["text"]," in grid 9")
        check()

def undo():#to undo the last move (saved in variable 'played' )
    global flag
    global turn
    global played
    if(played >=1 and played <=9):
        flag=flag-1
    if(played==1):
        played=0
        btn1["text"]=" "
        if turn==1:
            turn =2
        else:
            turn=1
    elif(played==2):
        played=0
        btn2["text"]=" "
        if turn==1:
            turn =2
        else:
            turn=1
    elif(played==3):
        played=0
        btn3["text"]=" "
        if turn==1:
            turn =2
        else:
            turn=1

```



```
elif(played==4):
    played=0
    btn4["text"]=" "
    if turn==1:
        turn =2
    else:
        turn=1
elif(played==5):
    played=0
    btn5["text"]=" "
    if turn==1:
        turn =2
    else:
        turn=1
elif(played==6):
    played=0
    btn6["text"]=" "
    if turn==1:
        turn =2
    else:
        turn=1
elif(played==7):
    played=0
    btn7["text"]=" "
    if turn==1:
        turn =2
    else:
        turn=1
elif(played==8):
    played=0
    btn8["text"]=" "
    if turn==1:
        turn =2
    else:
        turn=1
elif(played==9):
    played=0
    btn9["text"]=" "
    if turn==1:
        turn =2
    else:
        turn=1
```

```

else:
    print("Only 1 undo is allowed")

def check():#checking for the winner
    global flag
    global turn
    b1 = btn1["text"]
    b2 = btn2["text"]
    b3 = btn3["text"]
    b4 = btn4["text"]
    b5 = btn5["text"]
    b6 = btn6["text"]
    b7 = btn7["text"]
    b8 = btn8["text"]
    b9 = btn9["text"]
    flag=flag+1

    def min1():
        if b1==b2 and b1==b3 and b1=="O" or b1==b2 and b1==b3 and b1=="X":
            win(btn1["text"])
    def min2():
        if b4==b5 and b4==b6 and b4=="O" or b4==b5 and b4==b6 and b4=="X":
            win(btn4["text"])
    def min3():
        if b7==b8 and b7==b9 and b7=="O" or b7==b8 and b7==b9 and b7=="X":
            win(btn7["text"])
    def min4():
        if b1==b4 and b1==b7 and b1=="O" or b1==b4 and b1==b7 and b1=="X":
            win(btn1["text"])
    def min5():
        if b2==b5 and b2==b8 and b2=="O" or b2==b5 and b2==b8 and b2=="X":
            win(btn2["text"])
    def min6():
        if b3==b6 and b3==b9 and b3=="O" or b3==b6 and b3==b9 and b3=="X":
            win(btn3["text"])
    def min7():
        if b1==b5 and b1==b9 and b1=="O" or b1==b5 and b1==b9 and b1=="X":
            win(btn1["text"])
    def min8():
        if b7==b5 and b7==b3 and b7=="O" or b7==b5 and b7==b3 and b7=="X":
            win(btn7["text"])

```

```

#multithreading - 8 threads - to check for all patterns(8) of winning the
game

t1 = threading.Thread(target=min1)
t2 = threading.Thread(target=min2)
t3 = threading.Thread(target=min3)
t4 = threading.Thread(target=min4)
t5 = threading.Thread(target=min5)
t6 = threading.Thread(target=min6)
t7 = threading.Thread(target=min7)
t8 = threading.Thread(target=min8)

t1.start()
t2.start()
t3.start()
t4.start()
t5.start()
t6.start()
t7.start()
t8.start()

if flag ==10:
    messagebox.showinfo("Tie", "Match Tied!!! Try again :)")
    print("Match over!")
    print("Match tied")
    turn=1
    flag=1
    window.destroy()

#saving the winner details
def win(player):
    global turn
    global flag
    file = open("stat.txt","a")
    print("The game is over !")
    if player=="X":
        ans = "Game complete " +name3 + " wins "
        winner= "PVP: "+name3+" wins\n"
        print(name3," wins")
    else :
        ans = "Game complete " +name4 + " wins "
        winner= "PVP: "+name4+" wins\n"
        print(name4," wins")
    file.write(winner)

```

```

        messagebox.showinfo("Congratulations", ans)
        turn=1
        flag=1
        window.destroy() # close the program

#displaying tick-tac-toe game grid
    btn1 = Button(window, text=" ",bg="black",
fg="white",width=3,height=2,font=('Roman','20'),command=clicked1)
    btn1.grid(column=1, row=7)
    btn2 = Button(window, text=" ",bg="black",
fg="white",width=3,height=2,font=('Roman','20'),command=clicked2)
    btn2.grid(column=2, row=7)
    btn3 = Button(window, text=" ",bg="black",
fg="white",width=3,height=2,font=('Roman','20'),command=clicked3)
    btn3.grid(column=3, row=7)
    btn4 = Button(window, text=" ",bg="black",
fg="white",width=3,height=2,font=('Roman','20'),command=clicked4)
    btn4.grid(column=1, row=8)
    btn5 = Button(window, text=" ",bg="black",
fg="white",width=3,height=2,font=('Roman','20'),command=clicked5)
    btn5.grid(column=2, row=8)
    btn6 = Button(window, text=" ",bg="black",
fg="white",width=3,height=2,font=('Roman','20'),command=clicked6)
    btn6.grid(column=3, row=8)
    btn7 = Button(window, text=" ",bg="black",
fg="white",width=3,height=2,font=('Roman','20'),command=clicked7)
    btn7.grid(column=1, row=9)
    btn8 = Button(window, text=" ",bg="black",
fg="white",width=3,height=2,font=('Roman','20'),command=clicked8)
    btn8.grid(column=2, row=9)
    btn9 = Button(window, text=" ",bg="black",
fg="white",width=3,height=2,font=('Roman','20'),command=clicked9)
    btn9.grid(column=3, row=9)

undo=Button(window,text="Undo",bg="yellow",fg="black",width=3,height=1,font=('Times
','20'),command=undo)
    undo.grid(column=2,row=10)
    window.mainloop()

#displaying the welcome screen - to choose the game
button1 =Button(root, text ="Player Vs
Player",font=('Times','12','bold'),bg='yellow', command =info1) #command linked

```

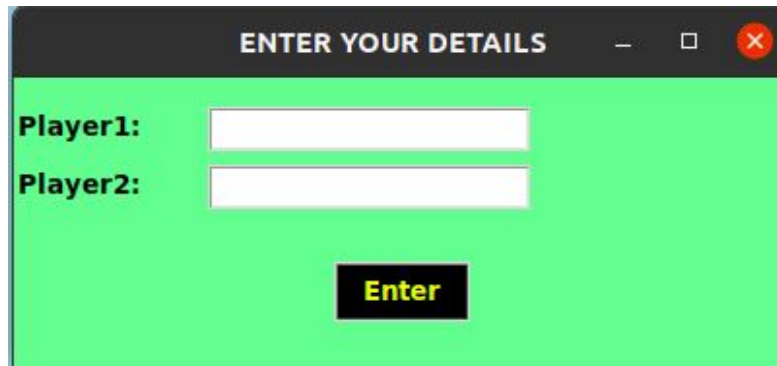
```
button1.pack()
button1.place(relx=0.5, rely=0.3, anchor=CENTER)

button2 = Button(root, text="Player Vs
Computer", font=('Times', '12', 'bold'), bg='yellow', command =info2) #create a new
function
button2.pack()
button2.place(relx=0.5, rely=0.45, anchor=CENTER)

button3
=Button(root, text="Statistics", font=('Times', '12', 'bold'), bg='yellow', command
=open_stat)
button3.pack()
button3.place(relx=0.5, rely=0.6, anchor=CENTER)
root.configure(bg='lightblue')
root.mainloop()
```

Player Vs Player :

- In our player vs player scenario when we click on player Vs player button the program control transfers to “**info1**” function block.
- There the program asks to enter the player names and once the player names are entered it assigns X and O to both the players.



- The tic tac toe boxes displayed to play are nothing but buttons using tkinter.
- **turn** and **flag** variables are declared as global variables. Flag indicates the number of moves played in the match(helps to check if there is a tie in the match) and turn indicates which player's turn it is to play.
- Initially both turn and flag are set to 1
- Player 1 is assigned 'X' and player two is assigned 'O' all times.
- Now as turn is 1 initially , the player 1 gets turn to mark in the box .
- When a player clicks on any box say n ($0 < n < 10$ as shown in below figure), the program control will be transferred to the **clicked 'n'** function.

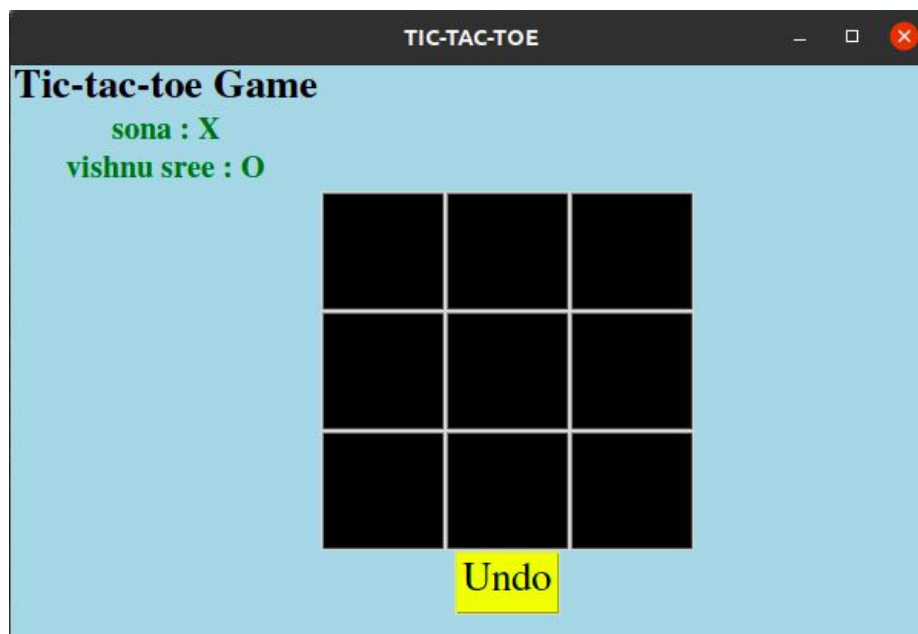
1	2	3
4	5	6
7	8	9

- In the **clicked 'n'** function we first check if that particular box is not marked before. If it is not marked then we check the turn variable. If turn is 1 we enter X into that box and make the turn 2, if turn is 2 we enter O into the box and make turn 1.
- In this way after we enter the value into the box we call the **check** function to see if that player has won.
- Each time the **check** function is called we increment the value of **flag** by 1.
- In the **check** function we have made **8 function** definitions which represents all the eight possibilities of winning (3 rows + 3 columns + 2 diagonals). Each of these 8 checks if all the

three values in row or column or diagonal are either X or O i.e, 1st function checks if all 3 values in first row are either X or O , similarly 2nd function checks if all 3 values in second row are either X or O and so on.

- In the **check** function we created **8 threads** and each thread is used to call each function from these 8 functions.
- If any one of these functions satisfy the condition that all 3 values are same , then we call **win()** function and pass the value in the block i.e, X or O
- In the **win()** function, we show a message that “player 1 wins” if passed parameter is X or it shows message “player 2 wins” if O is the passed parameter. It also opens a text file called **stat.txt** in append mode and saves that winner's details. This function sets back the value of the flag and turn to 1.
- This list of winners is displayed when we click on **statistics**
- In this process if the **check** function is called 9 times then it means that all the 9 boxes are filled and whenever the check function is called the **flag** is incremented as mentioned earlier. So if at any point of time the **flag reaches 10** it means all boxes are filled and if there is no winner ,then the program displays that the game has been tied.
- UNDO option:

As said earlier, whenever the player selects some box the control is shifted to the **clicked 'n'** function. There the **played** variable is set to 'n'. So if we press the undo button the value of the **played** variable is checked and the value in that box is erased . the **turn** will be changed to the previous players turn



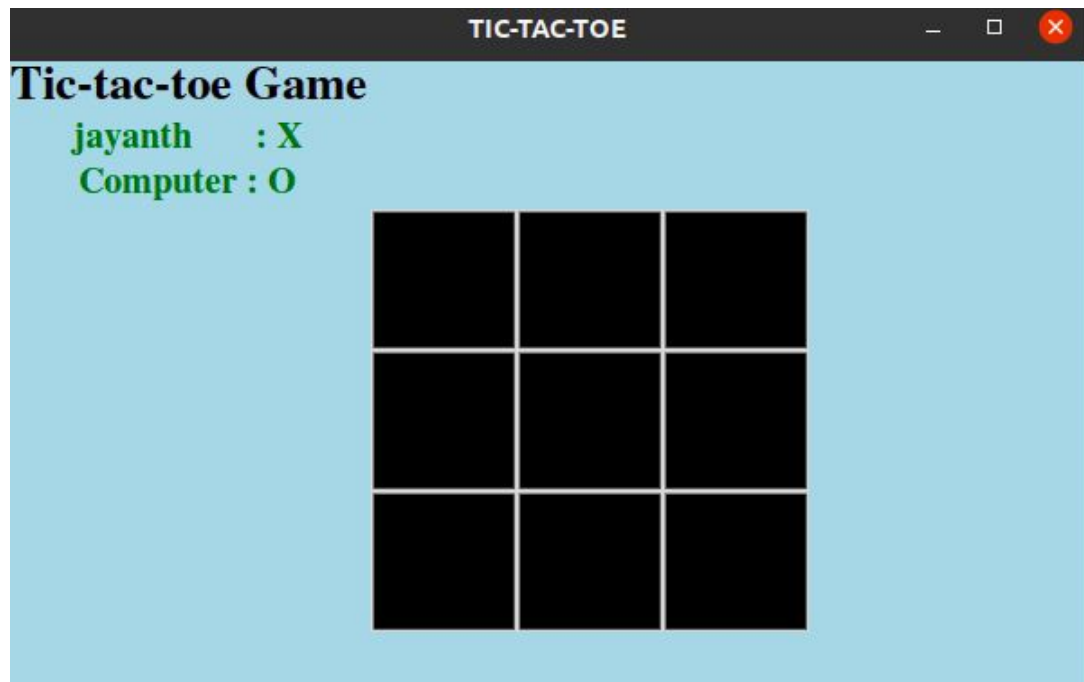
Player Vs Computer :

- In this we have the same functions as we have seen in Player V/s Player except that instead of player 2 we write an code in the program itself for the computer to decide which block it has to mark in.
- After entering the name of the player player is always assigned X and is given first turn while computer is given second turn and is assigned O



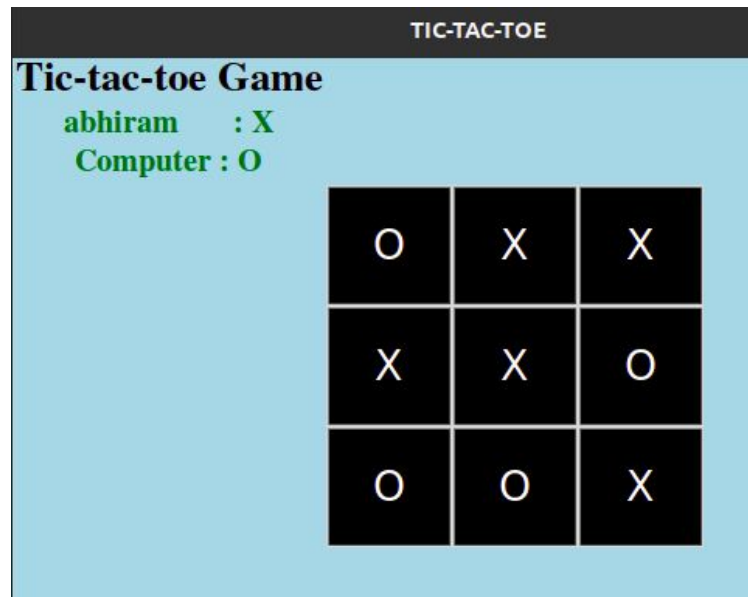
- When the computer gets the turn, it calls the function **maximo()**. Unlike the player v/s player scenario, the board status is saved in a list named `board[]` to find the best possible possible. By default, the 'board' is stored with its box numbers ie, [1,2,3,4,5,6,7,8,9]
- The maximo function creates a variable named `move` for storing the best move(box number) for the scenario. The move is set to -1 defaultly.
- **maximo()** looks at different possible scenarios for the computer to win by iterating through all the values from 1 to 9 and calling the **make_move()** to ensure it's the correct move.
- If none of the scenarios ensures a win for the computer, it looks for the moves that can prevent the player from winning.
- If it is still unable to find a specific move, we choose positions which are empty in the game. It first looks for the middle box (5), because it gives a higher chance of winning. If not available, it then looks for the corner boxes (1,7,3,9). Then looks for the left out ones (2,4,6,8).
- **maximo()** function returns the 'move' value
- **make_move()** function checks whether the given move can be made (ie, if the box is empty) by calling the **can_move()** function. Then checks whether the move will lead to the winning of the given player by calling the **can_win()** function. If the undo parameter passed is set to true, then the board value for the corresponding move position is restored i.e., say we want to undo the box 3 then `board[3-1]=3-1` is set as per the default initialization.(Note: index of the board starts from zero hence, the above subtraction).
- **make_move()** returns if the move is possible and whether it's a win.
- **can_move()** function checks if the value in the board corresponding to the 'move' value is the default value (by default, the 'board' is stored with its box numbers ie, [1,2,3,4,5,6,7,8,9]). If yes, then it returns TRUE else returns FALSE.

- **can_win()** function checks whether the corresponding move leads to winning of the given player. By checking the board with the 8 possible scenarios. It returns TRUE, if it's a win or else it returns FALSE.



Output Screenshots:

- Tied



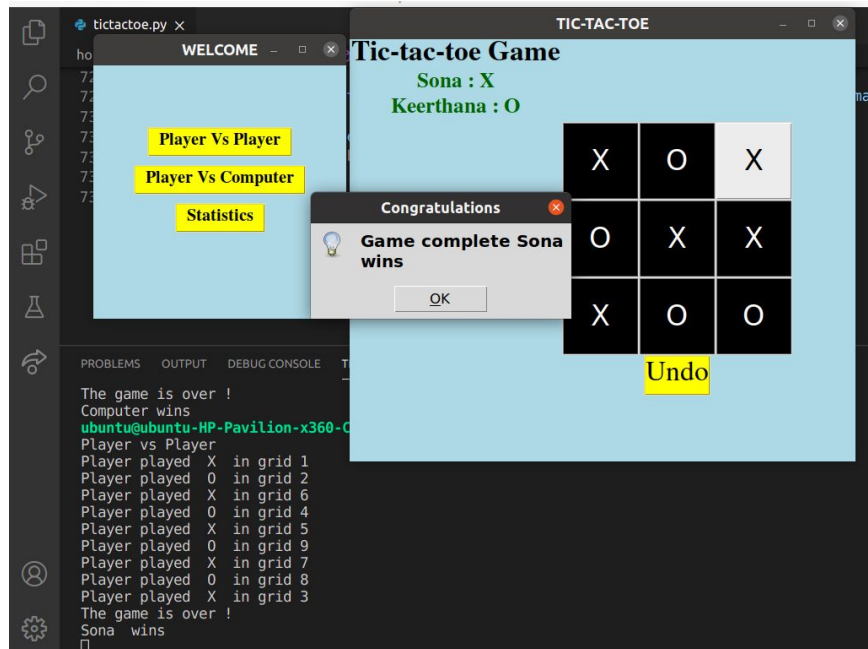
```
keerthi@keerthi:~/OS/pro/Update$ python3 tictactoe.py
Player vs Computer
Player played X in grid 5
Computer's turn: Plays grid 1
Player played X in grid 3
Computer's turn: Plays grid 7
Player played X in grid 4
Computer's turn: Plays grid 6
Player played X in grid 2
Computer's turn: Plays grid 8
Player played X in grid 9
The game is over !
Match tied
```

- Statistics:

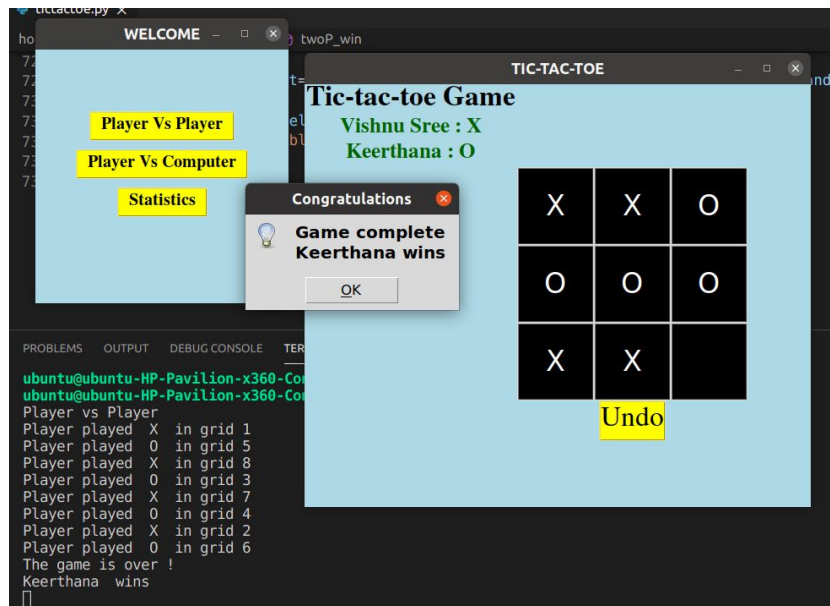
```
PVP: keerthana wins
PVP: jayanth wins
PVC: Computer wins
```

- Player v/s Player

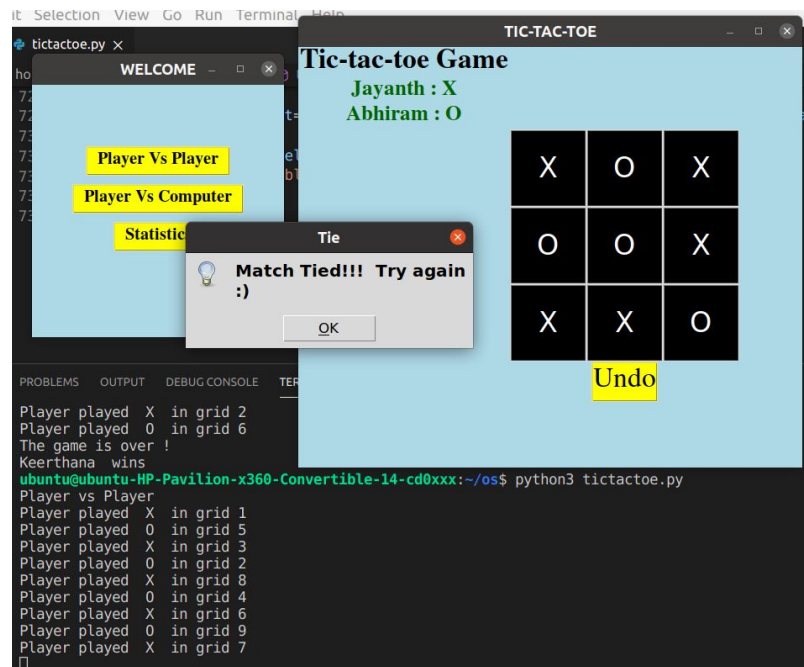
- ❑ Player 1 wins



- ❑ Player 2 wins

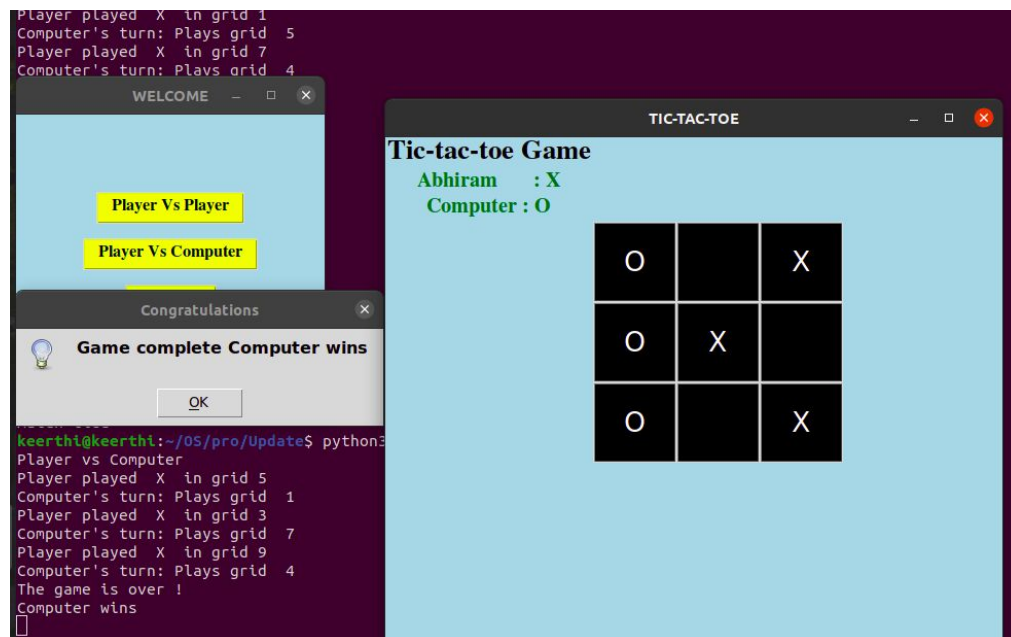


❏ Draw



● Player v/s Computer

❏ Computer wins



Draw

