TIC TAC TOE

SOCKET PROGRAMMING

What is our Project about?

CN

TIC TAC TOE

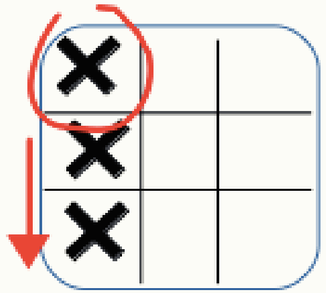
ALSO KNOWN AS NOUGHTS AND CROSSES

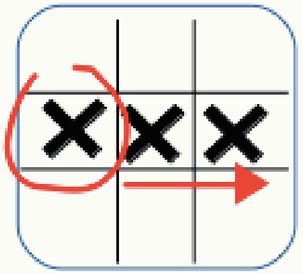
Very popular children’s pencil and paper board game, which is often played and enjoyed by many adults, as well. Because of its simplicity, this 3 row per 3 row board game may seem trivial at first, however, Tic Tac Toe involves its share of analytics and rapidity.

GAME RULES

Checking the win conditions

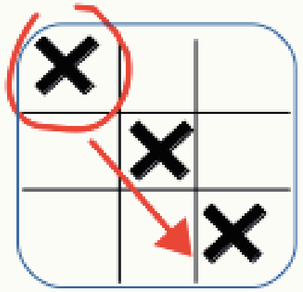
topmost cell
vertical strike

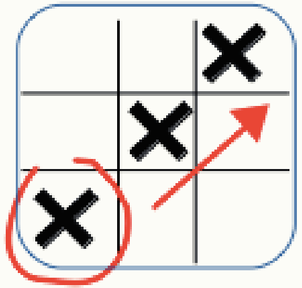




leftmost cell
horizontal strike

top leftmost cell
left diagonal strike





bottom leftmost cell
right diagonal strike

HOW TO PLAY

01 Two players can play with each other by using a 3×3 board in Tic Tac Toe.

02 One player chooses noughts, the opposing player uses crosses

03 The first player to align 3 of their identical symbols, either noughts or crosses, (horizontally, vertically or diagonally) wins the game.

Steps Involved

Socket

Send

Check

Result

Design

STEP-1 : SOCKET

SERVER

Initialising socket:

```
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

Setting required options:

```
 s.setsockopt(socket.SOL_SOCKET,socket.SO_REUSEADDR,1)
```

Declaring host and port:

```
 host = '127.0.0.1'
 port = 8900
```

Binding host and port:

```
 s.bind((host,port))
```

Waiting for client to connect:

```
 s.listen(5)
```

Accepting client's connection:

```
c,ad = s.accept()
```

CLIENT

Initialising Socket:

```
c = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

Setting required options:

```
c.setsockopt(socket.SOL_SOCKET,socket.SO_REUSEADDR,1)
```

Declaring host and port:

```
host = '127.0.0.1'
port = 8900
```

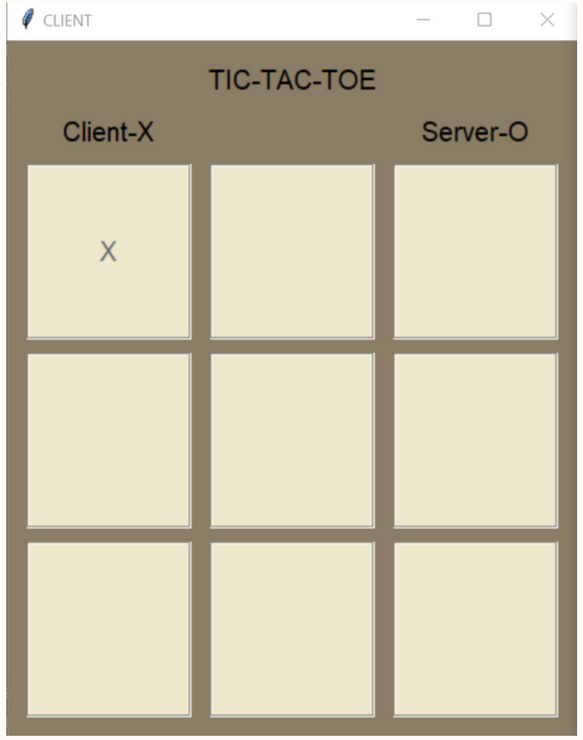
Connecting to server:

```
c.connect((host,port))
```

STEP-2 : Send

We initialise serverTurn as 0 and clientTurn as 1

We have 9 cells and to uniquley identify the cells, used ascii values from 'a'-'i'



When serverTurn is 1, we change it to 0 and send the message to server to display it there, and disable the cell.

```
def sendbtn1 ():
    global clientTurn
    if clientTurn == 1:
        clientTurn = 0
        msg = 'a'
        client.send(msg.encode('ascii'))
        btn1['text']='X'
        btn1['state']= tkinter.DISABLED
        global counter
        counter += 1
        check()
```

So, If 1st cell is clicked and serverTurn is 0, we display 'X' and disable the cell.
We change serverTurn to 1 to continue the game.


```
msg= c.recv(2048).decode('ascii')
if ( msg == 'a' and serverTurn == 0 ):
        btn1['text']='X'
        btn1['state']= tkinter.DISABLED
        serverTurn = 1
```

Similarly, for each cell when clicked.

STEP-3 : Check

We check positions and send win is it satisfies the conditions

```
def check():
    if(btn1['text'] == 'X'):
        if(btn2['text']=='X'):
            if(btn3['text']=='X'):
                sendWinner('X-Winner')
```

Similarly for positions:
(1,2,3) ; (4,5,6) ; (7,8,9)
(1,4,7) ; (2,5,8) ; (3,6,9)
(1,5,9) ; (3,5,7)

We declare a counter variable to check if game is tie

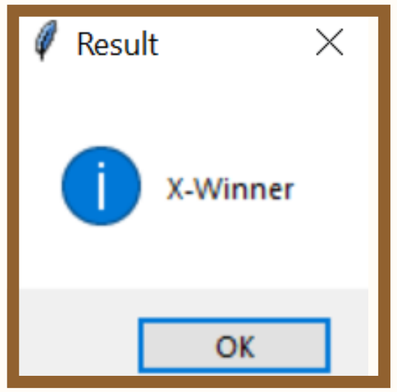
```
    if(counter == 5):
            sendWinner('It\'s a Tie')
```

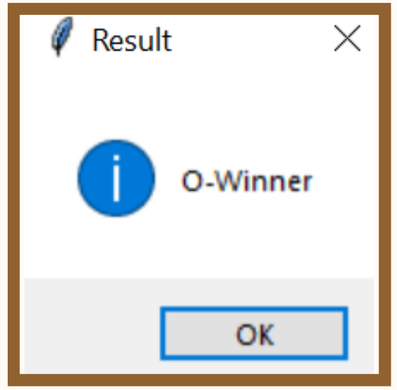
STEP-4 : Result

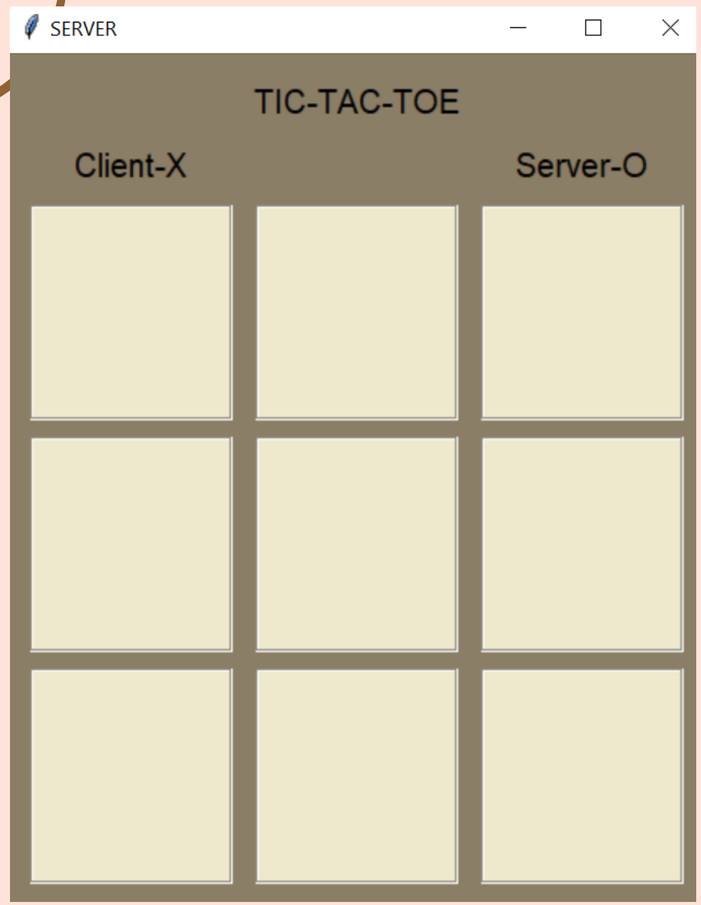
A message box is displayed with results like:

- X-Winner
- O- Winner
- It's a tie

After message box is closed, all the cells will be reset and we can play game again.







STEP - 5

Design

We used Tkinter module for designing server and clinet borads.

Libraries Used

Socket

Tkinter

_thread

MessageBox

CN

Presented By

K. VIDYA SREE

AM.EN.U4CSE18325

T. SATYA PRABHASA

AM.EN.U4CSE18356

B. RAJKUMAR

AM.EM.U4CSE18310

R. SAI RUTHWIK

AM.EN.U4CSE18345

THANK YOU !!