

## Algorithm for Tic Tac Toe

Tic Tac Toe Algorithm ( ) {

// Input:-

// Output:-

Step 1:- Create board

Initialize board as  $3 \times 3$  3D array  
with empty value

```
int [ ] [ ] board = new int [ 3 ][ 3 ]
for (int i=0; i<3; i++){
    for (int j=0; j<3; j++){
        board [i] [j] = 0;
    }
}
```

Step 2:-

Set user player = "x" & AI = "o";

S3:-

while Game not over || s

if player turn [

Display Board. ask player for  
input

if board [row] [column] is empty  
board [row] [column] = 'x'

else

ask for another input

}

III) turn

: if board [row] [column] is empty  
board [row] [column] = 'o'

if first move is S

else if player is suppose to win  
block the next player

else

place random of some column

: if board [row] [column] is empty  
board [row] [column] = 'o'

Step

IV) game over condition

: if board [row] [1, 2, 3] = 'x'  
X win;

else if board [1, 2, 3] [column] =  
'x'

player win;

else if diagonal = 'x'

player win;

else if board [row] = 'o'

S T win;

else if board [column] = 'o'

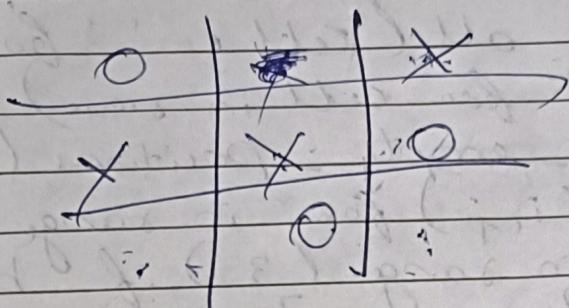
S T win;

else if diagonal, board [] = 'o'

S T win

else

draw



### Code

~~import random~~

```
def print_board(board):  
    for row in board:
```

```
        print (" ".join(row))  
        print ("--- * ---")
```

```
def check_winner(board):  
    lines = []
```

```
    lines.append(board)
```

```
    lines.append([board[i][j] for i  
                in range(3) for j in range(3)])
```

```
    lines.append([board[i][j] for i  
                in range(3)])
```

```
    lines.append([board[i][2-i] for i  
                in range(3)])
```

```
    for line in lines:
```

```
        if line[0] == line[1] == line[2]:
```

```
            return line[0]
```

```
    return None
```

```
return None
```

Date \_\_\_\_\_  
Page \_\_\_\_\_

```
def is_board_full(board):
    # Iterates over all cells in the board
    for row in board:
        for cell in row:
            if cell != ' ':
                return False
    return True

def get_empty_position(board):
    # Iterates over cells in the board
    for i in range(3):
        for j in range(3):
            if board[i][j] == ' ':
                return [i, j]
    return None
```

```
def player_move(board):
    white = True
    try:
        row = int(input("Enter row (0 - 2): "))
        col = int(input("Enter column (0 - 2): "))
        if board[row][col] != ' ':
            print("Position already taken.\nTry again.")
            raise ValueError
        board[row][col] = 'X' if white else 'O'
        break
    except ValueError:
        print("Invalid input. Please enter numbers between 0 and 2")
```

```
def ai_move(board):
    for row, col in get_empty_position(board):
        board[row][col] = 'O'
        if check_winner(board) == 'O':
            return [row, col]
    return None
```

for row, col in get\_empty\_positions  
(board):

board[row][col] = 'x'

if check\_winner(board) == 'x':

board[row][col] = 'o'

return

board[row][col] = ''

row, col = random.choice(get\_empty\_positions(board))

board[row][col] = 'o'

def main():

board = [[ 'x' if i in range(3)  
for i in range(3)]]

current\_player = 'x'

while True:

print\_board(board)

if current\_player == 'x':

player\_move(board)

else:

ai\_move(board)

winner = check\_winner(board)

if winner:

print\_board(board)

print(f'{winner} wins!')

break

elif is\_board\_full(board):

print\_board(board)

Point ("It's a draw!")  
break

current-player = 'O' if current  
player == 'X' else 'X'

if name == "main":  
 main()

Output

  |  |  
  ---

  |  |  
  ---

  |  |  
  ---

Enter row (0-2): 1

Enter column (0-2): 1

(board) board

  |  |  
  ---

  |  |  
  ---

  |  |  
  ---

  |  |  
  ---

Enter row (0-2): 0

Enter column (0-2): 0

X  |  |  
  ---

  |  |  
  ---

  |  |  
  ---

$$\begin{array}{r} \cancel{1} \cancel{x} \cancel{1} \\ - \\ 0 1 1 0 \end{array}$$

Enter row ( $0 - 2$ ): 2

Enter column ( $0 - 2$ ): 1

$$\begin{array}{r} \cancel{x} \cancel{1} \cancel{1} \\ - \\ \cancel{1} \cancel{x} \cancel{1} \\ - \\ 0 1 \cancel{x} \cancel{1} 0 \\ - \\ \cancel{x} 1 0 \ 1 \\ - \\ \cancel{1} \cancel{x} \cancel{1} \\ - \\ 0 \ 1 \cancel{x} \cancel{1} 0 \\ - \end{array}$$

Enter row ( $0 - 2$ ): 1

Enter column ( $0 - 2$ ): 2

$$\begin{array}{r} \cancel{x} 1 0 \ 1 \\ - \\ \cancel{1} \cancel{x} \cancel{1} \cancel{x} \\ - \\ 0 \ 1 \cancel{x} \cancel{1} 0 \\ - \\ \cancel{x} 1 0 \ 1 \\ - \\ \cancel{0} \cancel{1} \cancel{x} \cancel{1} \cancel{x} \\ - \\ 0 \ 1 \cancel{x} \cancel{1} 0 \end{array}$$

Enter row ( $0 - 2$ ): 0

Enter column ( $0 - 2$ ): 2

$$\begin{array}{r} \cancel{x} \cancel{1} 0 \ 1 \cancel{x} \\ - \\ 0 \ 1 \cancel{x} \cancel{1} \cancel{x} \\ - \\ 0 \ 1 \cancel{x} \cancel{1} 0 \end{array}$$

It's a draw!

*Sel*