

## 1. Tic Tac Toe (user v/s computer).

→ Algorithm:

Step 1: Randomize (with user starts first if computer else then randomize "some" marking and assign  $\leftarrow x$ .)

user will (pick) a space instead of  $x$ .

Step 2: (i) Second player [get input]  $\rightarrow$  if user  $\neq ! [s]$

Step 3: Construct (a) play matrix of  $3 \times 3$  matrix.

$\text{board} = [0][0] \text{ board} = [0][1] \text{ board} = [0][2]$

Step 4: First player marks a! space on the board.

$[0][0] \text{ board}$

Step 5: Every turn alternatively switches until one

= 2 players gets a = horizontal, diagonal or vertical

: rows / columns filled with the same symbol

$x \parallel y$

$[0][0] \text{ board} = [1][0] \text{ board} = [0][1] \text{ board} \dots$

Step 6: Conditions: "in a tic tac toe that can occur to determine" but more: when

- i) get three in a row - win when
- ii) Block an almost complete sequence - Block
- iii) opportunity to winning two ways. etc

Step 7: The game ends with either of the players winning or resulting in a Tie.

Code :

```
import random
```

```
def print_board(board):
    for row in board:
        print(" ".join(row))
```

```
def check_winner(board):
    for i in range(3):
        if board[0][i] == board[1][i] == board[2][i] != " ":
            return board[0][i]
        if board[0][i] == board[1][i] == board[2][i] != " ":
            return board[0][i]
    if board[0][0] == board[1][1] == board[2][2] != " ":
        return board[0][0]
```

```
def is_board_full(board):
    for row in board:
        for cell in row:
            if cell != " ":
```

```
def ai_name(board):  
    for i in range(3):  
        for j in range(3):  
            if board[i][j] == " ":  
                board[i][j] = "O".  
                if check_winner(board) == "O":  
                    return  
                board[i][j] = " "
```

```
for i in range(3):  
    for j in range(3):  
        if board[i][j] == " ":  
            board[i][j] = "X".  
            if check_winner(board) == "X":  
                board[i][j] = "O"  
                return  
            board[i][j] = " "
```

```
if board[4][1] == " ":  
    board[4][1] = "O"  
    return
```

corner = [(0,0), (0,2), (2,0), (2,2)]

random.shuffle(corners)

for corner in corners:

```
if board[corner[0]][corner[1]] == " ":  
    board[corner[0]][corner[1]] = "O"  
    return
```

sides = [(0,1), (1,0), (2,2), (2,1)]

random.shuffle(sides)

pause

home

end

pg

p

for side in sides:

if board [side[0] [side[1]]] == " ":

board [side[0] [side[1]]] = "O"

return

def play\_game ():

board = [ [" " for \_ in range(3) ] for \_ in range(3) ]  
print (" welcome to Tic Tac Toe game")

choice = int (input ("Enter 1 or 0 for header & 0 for  
Tails"))

x = random . randint (0,1)

if choice == x:

print (" You have won the toss ! ")

player\_fist = True

else:

print (" AI won the toss ! AI goes first")

player\_fist = False

print\_board (board)

while True:

if player\_fist == 0: (0,0) = X

while True: (0,0) = X

try:

row = int (input (" Enter row (1-3) : ")) - 1

column = int (input (" Enter column (1-3) : ")) - 1

if board [row] [column] == " ":

board [row] [column] = "X"

break

else:

```
print (" all already taken, chose another ")
except ( ValueError, IndexError ):
    print (" Invalid input. Please enter between 1 and 3 ")
print_board (board)

if check_winner (board) == "X":
    print (" You win ! ")
    break
if is_board_full (board):
    print (" It is a draw ! ")
    break
player_first = False
else:
    # AI move
    print (" AI's turn . . . ")
    ai_move (board)
    print_board (board)

if check_winner (board) == "O":
    print (" AI wins ! ")
    break
if is_board_full (board):
    print (" Its a draw ")
    break

player_first = True

if __name__ == "__main__":
    play_game ()
```

(Output) Select 1 or 0:

You have won the toss!

~~(Input) Enter row~~

Enter row (1-3) : 1

Enter column (1-3) & 1: 0

AI's turn

~~(Input) Enter column~~

Enter row (1-3): 1

" col (1-3): 1

~~(Input) Enter row~~

X | X | X | X | O | X

~~(Input) Enter column~~

~~(Input) Enter column~~

~~(Input) Enter column~~

X | O | X |  
O | O | X |  
X | X | O |

Is it a draw?

Welcome to Tic Tac Toe!

 |  |

-----

 |  |

-----

 |  |

-----

Enter row (1-3): 2

Enter column (1-3): 1

 |  |

-----

x |  |

-----

 |  |

-----

AI's turn...

 |  |

-----

x | o |

-----

 |  |

-----

Enter row (1-3): 1

Enter column (1-3): 1

x |  |

-----

x | o |

-----

 |  |

-----

AI's turn...

x |  |

-----

x | o |

-----

o |  |

-----

Enter row (1-3): 1  
Enter column (1-3): 3

X | | X

-----

X | O |

-----

O | |

-----

AI's turn...

X | O | X

-----

X | O |

-----

O | |

-----

Enter row (1-3): 3

Enter column (1-3): 2

X | O | X

-----

X | O |

-----

O | X |

-----

AI's turn...

X | O | X

-----

X | O |

-----

O | X | O

-----

Enter row (1-3): 2

Enter column (1-3): 3

X | O | X

-----

X | O | X

-----

O | X | O

-----

It's a draw!

```
-----
Enter row (1-3): 1
Enter column (1-3): 2
X | X |
-----
| O |
-----
|   |
-----
AI's turn...
X | X | O
-----
| O |
-----
|   |
-----
Enter row (1-3): 1
Enter column (1-3): 2
Cell already taken, choose another.
Enter row (1-3): 3
Enter column (1-3): 1
X | X | O
-----
| O |
-----
X |   |
-----
AI's turn...
X | X | O
-----
O | O |
-----
X |   |
-----
Enter row (1-3): 3
Enter column (1-3): 2
X | X | O
-----
O | O |
-----
X | X |
-----
AI's turn...
X | X | O
-----
O | O | O
-----
X | X |
-----
AI wins!
```