

AI

Python Tic Tac Toe game

b

Algorithm

d* Starting algorithm *

- Create a character board (3x3)
- Call the random function between 0
- 0 for player and 1 for computer.

board = [

```
[', ', ''],  
[', ', ''],  
[', ', '']
```

]

a = math.random() * Math.random() * 1 + 1

if (a == 0) {

print("computer plays first")

} else {

print("user plays")

while (true) {

playerPlays(board)

checkWin(board)

player()

compPlays(board)

}

if (boardFull()) print("board full")

}

```

function checkWin(board):
    win_cases = [
        {(0, 0), (0, 1), (0, 2)},
        {(1, 0), (1, 1), (1, 2)},
        {(2, 0), (2, 1), (2, 2)},
        {(0, 0), (1, 0), (2, 0)},
        {(0, 1), (1, 1), (2, 1)},
        {(0, 2), (1, 2), (2, 2)},
        {(0, 0), (1, 1), (2, 2)},
        {(0, 2), (1, 1), (2, 0)}
    ]

```

for i in win-cases:
 check for matching pattern
 if found return true

else return false

X	X	O
O	O	O
+		

```

function playerPlays(board):
    print("Enter the position")
    a, b = input("Enter").split()
    board[a][b] = 'X'
    if (checkWin(board))
        print("Player won the game")
}

```

DATE: PAGE:
O X O X X O O
O X

function compPlays(board) {

cases = [
 {(0, 0), (0, 2)}, {(0, 0), (2, 2)},
 {(1, 0), (1, 2)}]

]

for i in cases:

 if (i is present in board for user){
 // mark in the middle
 board[a][b] = 'O'
 }

 if (i is present in board for comp){
 // mark in the middle
 board[a][b] = 'X'
 }

 else {

 // mark anywhere

}

}

Code

```
import random

board = [['', '', ''], ['', '', ''], ['', '', '']]

def start():
    print('Welcome to Tic Tac Toe')
    nand = random.randint(0, 1)
    if nand == 0
        print('Player plays 1st')
    else
        print('Computer plays 2nd')
        computerPlays()

while True:
    if nand == 0
        playerPlays()
        if checkWin('X'):
            print('Player won')
            return
        nand = 1
    else
        computerPlays()
        if checkWin('O'):
            print('Computer won')
            return
```

```
def checkwin(player):  
    for win in wins:  
        if all(board[x,y] == player  
              for x,y in win):  
            return True  
    return False
```

```
def playerPlays():  
    while True:  
        try:  
            a,b = map(int, input().split())  
            if board[a][b] == "":  
                board[a][b] = 'X'  
            else:  
                print("Cell already taken.  
                Try again.")
```

```
def computerPlays():
```

```
    move = minimax(board, 0)  
    board[move[0]][move[1]] = 'O'
```

```
def minimax(board, currentPlayer):  
    if checkwin('X'): return -1  
    elif checkwin('O'): return 1  
    elif boardFull(): return 0
```

if currentPlayer
best-score
best-move

start()

output

computer play

		O

O	X	O
	X	

0,1

O	X	O
X	X	O
		O

if current player == 'O'

best-score = -float('inf')

best-move = None

start()

output

computer plays first

