

```
1 #Niranjan.v.2024.aiml@rajalakshmi.edu.in
2 PLAYER_X = 1
3 PLAYER_O = -1
4 EMPTY = 0
5
6 def evaluate(board):
7     for row in range(3):
8         if board[row][0] == board[row][1] ==
board[row][2] != EMPTY:
9         return board[row][0]
10    for col in range(3):
11        if board[0][col] == board[1][col] ==
board[2][col] != EMPTY:
12        return board[0][col]
13    if board[0][0] == board[1][1] == board[2][2] !=
EMPTY:
14        return board[0][0]
15    if board[0][2] == board[1][1] == board[2][0] !=
EMPTY:
16        return board[0][2]
17    return 0
18
19 def isMovesLeft(board):
20     for row in range(3):
21         for col in range(3):
22             if board[row][col] == EMPTY:
23                 return True
24     return False
25
26 def minimax(board, isMax):
27     score = evaluate(board)
28     if score == PLAYER_X:
29         return score
30     if score == PLAYER_O:
31         return score
32     if not isMovesLeft(board):
33         return 0
34
```

```
35     if isMax:
36         best = -float('inf')
37         for row in range(3):
38             for col in range(3):
39                 if board[row][col] == EMPTY:
40                     board[row][col] = PLAYER_X
41                     best = max(best, minimax(board, not
isMax))
42                     board[row][col] = EMPTY
43             return best
44     else:
45         best = float('inf')
46         for row in range(3):
47             for col in range(3):
48                 if board[row][col] == EMPTY:
49                     board[row][col] = PLAYER_O
50                     best = min(best, minimax(board, not
isMax))
51                     board[row][col] = EMPTY
52             return best
53
54 def findBestMove(board):
55     bestVal = -float('inf')
56     bestMove = (-1, -1)
57
58     for row in range(3):
59         for col in range(3):
60             if board[row][col] == EMPTY:
61                 board[row][col] = PLAYER_X
62                 moveVal = minimax(board, False)
63                 board[row][col] = EMPTY
64
```

```
64
65         if moveVal > bestVal:
66             bestMove = (row, col)
67             bestVal = moveVal
68     return bestMove
69
70 def printBoard(board):
71     for row in board:
72         print(" ".join(["X" if x == PLAYER_X else "O" if x
73 == PLAYER_O else "." for x in row]))
74
75 board = [
76     [PLAYER_X, PLAYER_O, PLAYER_X],
77     [PLAYER_O, PLAYER_X, EMPTY],
78     [EMPTY, PLAYER_O, PLAYER_X]
79 ]
80
81 print("Current Board:")
82 printBoard(board)
83
84 move = findBestMove(board)
85 print(f"\nBest Move: {move}")
86
87 board[move[0]][move[1]] = PLAYER_X
88
89 print("\nBoard after best move:")
90 printBoard(board)
```

Current Board:

X	O	X
O	X	.
.	O	X

Best Move: (1, 2)

Board after best move:

X	O	X
O	X	X
.	O	X

[Program finished]