Game Theory---Finding The Best Move

```
function findBestMove(board):

    bestMove = NULL
    for each move in board :
        if current move is better than bestMove
            bestMove = current move
    return bestMove
```

```
function minimax(board, depth, isMaximizingPlayer):

    if current board state is a terminal state :
        return value of the board

    if isMaximizingPlayer :
        bestVal = -INFINITY

        for each move in board : value = minimax(board, depth+1, false)
            bestVal = max( bestVal, value)

            return bestVal
    else :
        bestVal = +INFINITY

        for each move in board :
            value = minimax(board, depth+1, true)
            bestVal = min( bestVal, value)

            return bestVal
```

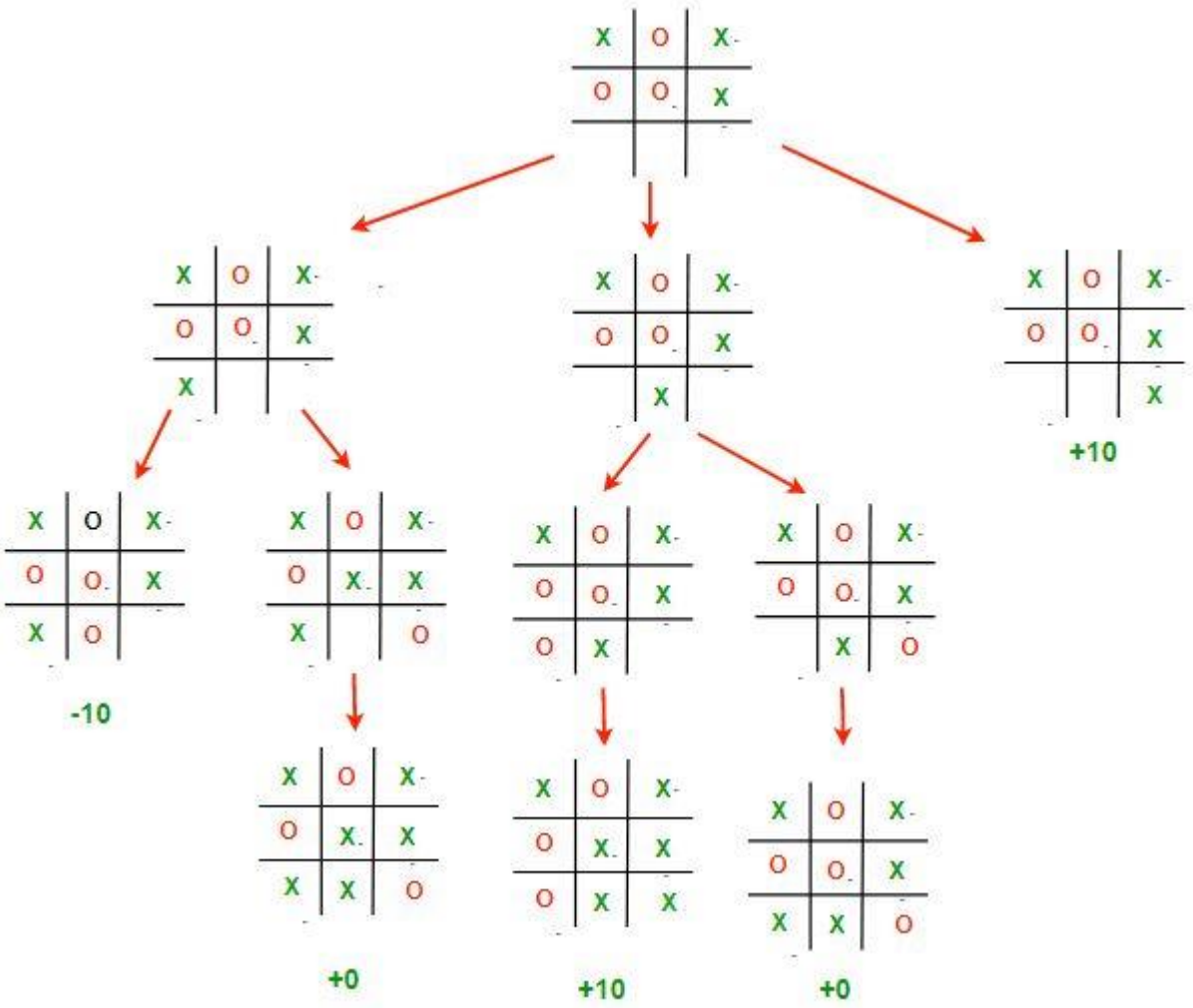
```
function isMovesLeft(board):

    for each cell in board:
        if current cell is empty:
            return true

    return false
```

```
if maximizer has won:
    return WIN_SCORE – depth
else if minimizer has won: r
    eturn LOOSE_SCORE + depth
```

Game Tree



The 3 possible scenarios in the above example are :

Left Move : If X plays [2,0]. Then O will play [2,1] and win the game. The value of this move is -10
Middle Move : If X plays [2,1]. Then O will play [2,2] which draws the game. The value of this move is 0
Right Move : If X plays [2,2]. Then he will win the game. The value of this move is +10;

Even though X has a possibility of winning if he plays the middle move, O will never let that happen and will choose to draw instead.