Sayantan Mukherjee 60009220131 D2-2

In [1]:
```
import math
def print_board(board):
    for row in board:
        print("|".join(row))
        print("-" * 5)
```

All winning Conditions Explored

In [2]:
```
def check_winner(board, player):
    win_conditions = [
        [board[0][0], board[0][1], board[0][2]],  # First row
        [board[1][0], board[1][1], board[1][2]],  # Second row
        [board[2][0], board[2][1], board[2][2]],  # Third row
        [board[0][0], board[1][0], board[2][0]],  # First column
        [board[0][1], board[1][1], board[2][1]],  # Second column
        [board[0][2], board[1][2], board[2][2]],  # Third column
        [board[0][0], board[1][1], board[2][2]],  # Diagonal left to right
        [board[0][2], board[1][1], board[2][0]],  # Diagonal right to left
    ]
    if [player, player, player] in win_conditions:
        return True
    return False
```

In [3]:
```
#Check if all squares are filled or not
def is_moves_left(board):
    for row in board:
        if ' ' in row:
            return True
    return False
```

Mini-Max Algorithm

In [4]:
```
def minimax(board, depth, is_max):
    if check_winner(board, 'O'):  # Computer wins
        return 10 - depth
    if check_winner(board, 'X'):  # Human wins
        return depth - 10
    if not is_moves_left(board):  # Tie
        return 0

    if is_max:
        best = -math.inf
        for i in range(3):
            for j in range(3):
                if board[i][j] == ' ':
                    board[i][j] = 'O'
                    best = max(best, minimax(board, depth + 1, not is_max))
```

```
                    board[i][j] = ' '
        return best
    else:
        best = math.inf
        for i in range(3):
            for j in range(3):
                if board[i][j] == ' ':
                    board[i][j] = 'X'
                    best = min(best, minimax(board, depth + 1, not is_max))
                    board[i][j] = ' '
        return best
```

Algo of Computer Play

In [5]:
```
def find_best_move(board):
    best_val = -math.inf
    best_move = (-1, -1)
    for i in range(3):
        for j in range(3):
            if board[i][j] == ' ':
                board[i][j] = 'O'
                move_val = minimax(board, 0, False)
                board[i][j] = ' '
                if move_val > best_val:
                    best_move = (i, j)
                    best_val = move_val
    return best_move
```

Main Game Function

In [6]:
```
def tic_tac_toe():
    board = [[' ' for _ in range(3)] for _ in range(3)]
    print("Tic-Tac-Toe Game! You (X) vs Computer (O)")
    print_board(board)

    while True:
        # Human move
        while True:
            row, col = map(int, input("Enter your move (row and column): ").
            if board[row][col] == ' ':
                board[row][col] = 'X'
                break
            else:
                print("Invalid move! Try again.")

        print_board(board)

        if check_winner(board, 'X'):
            print("Congratulations! You win!")
            break

        if not is_moves_left(board):
```

```
            print("It's a tie!")
            break

        # Computer move
        print("Computer is making a move...")
        row, col = find_best_move(board)
        board[row][col] = 'O'
        print_board(board)

        if check_winner(board, 'O'):
            print("Computer wins! Better luck next time.")
            break

        if not is_moves_left(board):
            print("It's a tie!")
            break
```

Let the Play Begin!

How to Play? Simple : Enter any number(0-2) in format for eg: 0 2 representing first row and Last Column and hopefully be victorious :)

In [10]:
```
tic_tac_toe()
```

```
Tic-Tac-Toe Game! You (X) vs Computer (O)
 | |
-----
 | |
-----
 | |
-----
Enter your move (row and column): 1 1
 | |
-----
 |X|
-----
 | |
-----
Computer is making a move...
O| |
-----
 |X|
-----
 | |
-----
Enter your move (row and column): 2 1
O| |
-----
 |X|
-----
 |X|
-----
Computer is making a move...
O|O|
-----
 |X|
-----
 |X|
-----
Enter your move (row and column): 0 2
O|O|X
-----
 |X|
-----
 |X|
-----
Computer is making a move...
O|O|X
-----
 |X|
-----
O|X|
-----
Enter your move (row and column): 1 0
O|O|X
-----
X|X|
-----
O|X|
-----
```

```
Computer is making a move...
O|O|X
-----
X|X|O
-----
O|X|
-----
Enter your move (row and column): 2 2
O|O|X
-----
X|X|O
-----
O|X|X
-----
It's a tie!
```