Roll: 20BCE514
AI PRACTICAL 8th
Aim: Min-Max Algorithm (Tic-Tac-Toe)

In [1]:

```
def MyGameBoard(gameBoard):
    print("Current State Of Board : \n\n")
    for i in range(0, 9):
        if (i > 0) and (i % 3) == 0:
            print("\n")
        if gameBoard[i] == 0:
            print("- ", end=" ")
        if gameBoard[i] == 1:
            print("O ", end=" ")
        if gameBoard[i] == -1:
            print("X ", end=" ")
    print("\n\n")
```

In [2]:

```
def player1Turn(tic_tac_toe):
    index = input("Enter X's position from [1...9]: ")
    index = int(index)
    if tic_tac_toe[index - 1] != 0:
        print("Wrong Move!!!")
        exit(0)
    tic_tac_toe[index - 1] = -1

def player2Turn(tic_tac_toe):
    index = input("Enter O's position from [1...9]: ")
    index = int(index)
    if (tic_tac_toe[index - 1] != 0):
        print("Wrong Move!!!")
        exit(0)
    tic_tac_toe[index - 1] = 1
```

In [3]:

```
def miniMaxAlgorithm(MyGameBoard, turn):
    x = checkTheBoard(MyGameBoard)
    if x != 0:
        return x * turn
    position = -1
    value = -2
    for i in range(0, 9):
        if MyGameBoard[i] == 0:
            MyGameBoard[i] = turn
            score = -miniMaxAlgorithm(MyGameBoard, (turn * -1))
            if score > value:
                value = score
                position = i
            MyGameBoard[i] = 0
    if position == -1:
        return 0
    return value
```

In [4]:

```
def CompTurn(board):
    index = -1
    value = -2
    for i in range(0, 9):
        if board[i] == 0:
            board[i] = 1
            score = -miniMaxAlgorithm(board, -1)
            board[i] = 0
            if score > value:
                value = score
                index = i
    board[index] = 1
```

In [5]:

```
def checkTheBoard(board):
    cb = [[0, 1, 2], [3, 4, 5], [6, 7, 8], [0, 3, 6], [1, 4, 7], [2, 5, 8], [0, 4, 8], [2,
    for i in range(0, 8):
        if (board[cb[i][0]] != 0 and board[cb[i][0]] == board[cb[i][1]] and board[cb[i][0]]
            return board[cb[i][2]]
    return 0
```

In [6]:

```
def main():
    choice = input("Enter 1 for single player, 2 for multiplayer: ")
    choice = int(choice)
    board = [0, 0, 0, 0, 0, 0, 0, 0, 0]
    if choice == 1:
        print("Computer : O Vs. You : X")
        player = input("Enter to play 1(st) or 2(nd) :")
        player = int(player)
        for i in range(0, 9):
            if checkTheBoard(board) != 0:
                break
            if (i + player) % 2 == 0:
                CompTurn(board)
            else:
                MyGameBoard(board)
                player1Turn(board)
    else:
        for i in range(0, 9):
            if checkTheBoard(board) != 0:
                break
            if i % 2 == 0:
                MyGameBoard(board)
                player1Turn(board)
            else:
                MyGameBoard(board)
                player2Turn(board)
    x = checkTheBoard(board)
    if x == 0:
        MyGameBoard(board)
        print("Draw!!!")
    if x == -1:
        MyGameBoard(board)
        print("X Wins!!! Y Loose !!!")
    if x == 1:
        MyGameBoard(board)
        print("X Loose!!! O Wins !!!!")
```

In [7]:

```
if __name__ == '__main__':
    main()
def player1Turn(tic_tac_toe):
    index = input("Enter X's position from [1...9]: ")
    index = int(index)
    if tic_tac_toe[index - 1] != 0:
        print("Wrong Move!!!")
        exit(0)
    tic_tac_toe[index - 1] = -1

def player2Turn(tic_tac_toe):
    index = input("Enter O's position from [1...9]: ")
    index = int(index)
    if (tic_tac_toe[index - 1] != 0):
        print("Wrong Move!!!")
        exit(0)
    tic_tac_toe[index - 1] = 1

def miniMaxAlgorithm(MyGameBoard, turn):
    x = checkTheBoard(MyGameBoard)
    if x != 0:
        return x * turn
    position = -1
    value = -2
    for i in range(0, 9):
        if MyGameBoard[i] == 0:
            MyGameBoard[i] = turn
            score = -miniMaxAlgorithm(MyGameBoard, (turn * -1))
            if score > value:
                value = score
                position = i
            MyGameBoard[i] = 0
    if position == -1:
        return 0
    return value

def CompTurn(board):
    index = -1
    value = -2
    for i in range(0, 9):
        if board[i] == 0:
            board[i] = 1
            score = -miniMaxAlgorithm(board, -1)
            board[i] = 0
            if score > value:
                value = score
                index = i
    board[index] = 1

def checkTheBoard(board):
    cb = [[0, 1, 2], [3, 4, 5], [6, 7, 8], [0, 3, 6], [1, 4, 7], [2, 5, 8], [0, 4, 8], [2,
    for i in range(0, 8):
        if (board[cb[i][0]] != 0 and board[cb[i][0]] == board[cb[i][1]] and board[cb[i][0]]
            return board[cb[i][2]]
    return 0

def main():
    choice = input("Enter 1 for single player, 2 for multiplayer: ")
    choice = int(choice)
    board = [0, 0, 0, 0, 0, 0, 0, 0, 0]
    if choice == 1:
        print("Computer : O Vs. You : X")
        player = input("Enter to play 1(st) or 2(nd) :")
        player = int(player)
        for i in range(0, 9):
            if checkTheBoard(board) != 0:
                break
            if (i + player) % 2 == 0:
                CompTurn(board)
            else:
                MyGameBoard(board)
                player1Turn(board)
    else:
        for i in range(0, 9):
            if checkTheBoard(board) != 0:
                break
            if i % 2 == 0:
                MyGameBoard(board)
                player1Turn(board)
            else:
                MyGameBoard(board)
                player2Turn(board)
    x = checkTheBoard(board)
    if x == 0:
        MyGameBoard(board)
        print("Draw!!!")
    if x == -1:
        MyGameBoard(board)
        print("X Wins!!! Y Loose !!!")
    if x == 1:
        MyGameBoard(board)
        print("X Loose!!! O Wins !!!!")

if __name__ == '__main__':
    main()
```

```
Enter 1 for single player, 2 for multiplayer: 1
Computer : O Vs. You : X
Enter to play 1(st) or 2(nd) :1
Current State Of Board :


-  -  -

-  -  -

-  -  -


Enter X's position from [1...9]: 1
Current State Of Board :


X  -  -
```

In [ ]: