

# **Indian Institute of Information Technology Surat**



## **Lab Report on Artificial Intelligence (CS 701) Practical**

**Submitted by**

**[RAHUL KUMAR SINGH] (UI21CS44)**

**Course Faculty**

**Dr. Ritesh Kumar**

**Mrs. Archana Balmik**

**Department of Computer Science and Engineering  
Indian Institute of Information Technology Surat  
Gujarat-394190, India**

**Aug-2024**

## Lab No: 10

### Aim:

Write a program for TIC – TAC –TOE game using min-max algorithm.

### Description:

- 3x3 board used for Tic-Tac-Toe game implementation.
- Minimax algorithm ensures AI makes optimal moves as 'O'.
- Player plays first as 'X' by selecting grid positions.
- Game ends when player wins, AI wins, or draw occurs.

### Code:

```
import math
def print_board(board):
    for row in board:
        print("| " + " | ".join(row) + " |")
        print("-" * 13)
def check_winner(board):
    for row in board:
        if row[0] == row[1] == row[2] != ' ':
            return row[0]
    for col in range(3):
        if board[0][col] == board[1][col] == board[2][col] != ' ':
            return board[0][col]
    if board[0][0] == board[1][1] == board[2][2] != ' ':
        return board[0][0]
    if board[0][2] == board[1][1] == board[2][0] != ' ':
        return board[0][2]
    return None
def is_draw(board):
    for row in board:
        if ' ' in row:
            return False
    return True
def minimax(board, depth, is_ai):
    winner = check_winner(board)
    if winner == 'X':
        return -1
    elif winner == 'O':
        return 1
    elif is_draw(board):
        return 0
    move = 'O' if is_ai else 'X'
    best_score = -math.inf if is_ai else math.inf
    func = None
    if is_ai:
        func = lambda x,y: max(x,y)
    else:
        func = lambda x,y: min(x,y)
    for i in range(3):
```

```

        for j in range(3):
            if board[i][j] == ' ':
                board[i][j] = move
                score = minimax(board, depth + 1, not is_ai)
                board[i][j] = ' '
                best_score = func(score, best_score)
    return best_score

def find_best_move(board):
    best_score = -math.inf
    move = None
    for i in range(3):
        for j in range(3):
            if board[i][j] == ' ':
                board[i][j] = 'O'
                score = minimax(board, 0, False)
                board[i][j] = ' '
                if score > best_score:
                    best_score = score
                    move = (i, j)
    return move

def main():
    board = [[' ' for _ in range(3)] for _ in range(3)]
    player_turn = True
    while True:
        print_board(board)
        if check_winner(board) == 'X':
            print("Player wins!")
            break
        elif check_winner(board) == 'O':
            print("AI wins!")
            break
        elif is_draw(board):
            print("It's a draw!")
            break
        if player_turn:
            row, col = map(int, input("Enter your move (row col): ").split())
            if board[row][col] == ' ':
                board[row][col] = 'X'
                player_turn = False
            else:
                print("Invalid move. Try again.")
        else:
            print("AI is making its move...")
            move = find_best_move(board)
            if move:
                board[move[0]][move[1]] = 'O'
                player_turn = True
if __name__ == "__main__":
    main()

```

**Output:**

```

PS D:\Assignment\CLASSROOM\Sem-7\AI\P10> python P10.py
| | | |
-----
| | | |
-----
| | | |
-----
Enter your move (row col): 0 0
| x | | |
-----
| | | |
-----
| | | |
-----
AI is making its move...
| x | | |
-----
| | o | |
-----
| | | |
-----
Enter your move (row col): 1 2
| x | | |
-----
| | o | x |
-----
| | | |
-----
AI is making its move...
| x | o | |
-----
| | o | x |
-----
| | | |
-----

```

```

Enter your move (row col): 2 1
| x | o | |
-----
| | o | x |
-----
| | x | |
-----
AI is making its move...
| x | o | |
-----
| | o | x |
-----
| o | x | |
-----
Enter your move (row col): 0 2
| x | o | x |
-----
| | o | x |
-----
| o | x | |
-----
AI is making its move...
| x | o | x |
-----
| | o | x |
-----
| o | x | o |
-----
Enter your move (row col): 2 0
Invalid move. Try again.
| x | o | x |
-----
| | o | x |
-----
| o | x | o |
-----

```

```

Enter your move (row col): 1 0
| x | o | x |
-----
| x | o | x |
-----
| o | x | o |
-----
It's a draw!

```

## Conclusion:

- AI always wins or results in a draw, player never wins.
- AI guarantees optimal play using Minimax for unbeatable strategy.
- Player's challenge lies in preventing AI from winning.