Min–Max Algorithm

The **Min–Max (or Minimax) algorithm** is a **decision-making algorithm used in two-player turn-based games** — especially games like **Tic-Tac-Toe, Chess, Checkers, Connect-4, etc.**

- One player is the **MAX player** (tries to maximize the score — wants to win)
- The other is the **MIN player** (tries to minimize the score — wants the MAX player to lose)

Min–Max is used by AI to **choose the best possible move**, assuming:
✓ Both opponents play **optimally**
✓ Both try to **win / avoid losing**

How Min–Max Works Step-by-Step

1. Generate all possible moves
2. For each move, simulate the opponent's best possible move
3. Continue until you reach a **terminal state** (win/lose/draw)
4. Assign a score to the terminal state
5. Backtrack and:

- MAX player picks the **highest score**
- MIN player picks the **lowest score**

6. The move that leads to the best score for MAX player becomes the chosen move

Example – When AI Plays in Tic-Tac-Toe

Suppose the AI is MAX and calculates the scores:

Move	**Outcome**	**Score**
A	AI wins	+10
B	Draw	0
C	AI loses	-10

Minimax will select **Move A** because it gives the **maximum score**.

Pseudocode

```
function minimax(node, depth, isMaximizingPlayer):
    if node is terminal:
        return score adjusted by depth

    if isMaximizingPlayer:
        best = -∞
        for each child of node:
            val = minimax(child, depth + 1, false)
            best = max(best, val)
        return best

    else:
        best = +∞
        for each child of node:
            val = minimax(child, depth + 1, true)
            best = min(best, val)
        return best
```

Minimax assumes:

✓ Opponent will not make mistakes

✓ Perfect and exhaustive calculation of future possibilities

Where It is Used in Games

You can apply minimax to create AI opponents in:

- Tic-Tac-Toe
- Connect-4
- Othello
- Chess (with improvements like Alpha-Beta pruning due to size)

To demonstrate Minimax clearly, the best example is:

Tic-Tac-Toe game where user plays against AI

The AI uses Minimax to decide its move and you can show:

- Generated scores for each move
- Highlighting which move was selected and why