

Minimax Algorithm

Here's a quick description of minimax. We are assuming "X" is always playing to maximize. Winning boards for X are scored as +1, for O as -1, draw as 0. Depth is there to limit the search depth. In tic-tac-toe you can search all the way to 9 (the whole board).

```
MINIMAX(board, depth, player):
    if game is over OR depth is 0:
        return score of board (+1 for X win, -1 for O win, 0 for draw)

    if player is X (maximizer):
        best = -infinity
        for each empty space:
            make the move
            best = max(best, MINIMAX(new board, depth-1, 0))
        return best

    if player is O (minimizer):
        best = +infinity
        for each empty space:
            make the move
            best = min(best, MINIMAX(new board, depth-1, X))
        return best

GET_MOVE(board, player):
    for each empty space:
        make the move
        score = MINIMAX(new board, max_search_depth, other player)
        keep track of the move with the best score
    return best move
```