```python
def print_board(board):
    for row in board:
        print(" | ".join(row))
        print("-" * 17)

def is_winner(board, player):
    # Check rows, columns, and diagonals for a win
    for i in range(3):
        if all(board[i][j] == player for j in range(3)) or all(board[j][i] == player for j in range(3)):
            return True
    if all(board[i][i] == player for i in range(3)) or all(board[i][2 - i] == player for i in range(3)):
        return True
    return False

def is_board_full(board):
    return all(board[i][j] != " " for i in range(3) for j in range(3))

def get_empty_cells(board):
    return [(i, j) for i in range(3) for j in range(3) if board[i][j] == " "]

def minimax(board, depth, maximizing_player):
    if is_winner(board, "O"):
        return 1
    elif is_winner(board, "X"):
        return -1
    elif is_board_full(board):
        return 0

    if maximizing_player:
        max_eval = float("-inf")
        for i, j in get_empty_cells(board):
            board[i][j] = "O"
            eval = minimax(board, depth + 1, False)
            board[i][j] = " "
            max_eval = max(max_eval, eval)
        return max_eval
    else:
```

```
38          min_eval = float("inf")
39          for i, j in get_empty_cells(board):
40              board[i][j] = "X"
41              eval = minimax(board, depth + 1, True)
42              board[i][j] = " "
43              min_eval = min(min_eval, eval)
44          return min_eval
45
46  def get_best_move(board):
47      best_move = None
48      best_eval = float("-inf")
49      for i, j in get_empty_cells(board):
50          board[i][j] = "O"
51          eval = minimax(board, 0, False)
52          board[i][j] = " "
53          if eval > best_eval:
54              best_eval = eval
55              best_move = (i, j)
56      return best_move
57
58  def play_game():
59      board = [[" " for _ in range(3)] for _ in range(3
    )]
60
61      while True:
62          print_board(board)
63
64          # Player's move
65          row = int(input("Enter the row (0, 1, or 2
    ): "))
66          col = int(input("Enter the column (0, 1, or 2
    ): "))
67          if board[row][col] == " ":
68              board[row][col] = "X"
69          else:
70              print("Invalid move. Try again.")
71              continue
72
73          if is_winner(board, "X"):
74              print_board(board)
75              print("You win!")
```

```
            break
        elif is_board_full(board):
            print_board(board)
            print("It's a draw!")
            break

        # AI's move
        print("AI is thinking...")
        ai_row, ai_col = get_best_move(board)
        board[ai_row][ai_col] = "O"

        if is_winner(board, "O"):
            print_board(board)
            print("AI wins!")
            break
        elif is_board_full(board):
            print_board(board)
            print("It's a draw!")
            break

if __name__ == "__main__":
    play_game()
```