```python
import math

PLAYER_X = 'X'  # AI
PLAYER_O = 'O'  # Human
EMPTY = ' '

def print_board(board):
    """Prints the Tic-Tac-Toe board."""
    for row in board:
        print(" | ".join(row))
        print("-" * 9)

def is_winner(board, player):
    """Checks if a player has won the game."""
    for row in board:
        if all(cell == player for cell in row):
            return True

    for col in range(3):
        if all(board[row][col] == player for row 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_draw(board):
    """Checks if the game is a draw."""
    return all(board[row][col] != EMPTY for row in range(3) for col in range(3))

def evaluate(board):
    """Evaluates the board state."""
    if is_winner(board, PLAYER_X):
        return 10
    if is_winner(board, PLAYER_O):
        return -10
    return 0

def minimax(board, depth, is_maximizing, alpha, beta):
    """Minimax algorithm with alpha-beta pruning."""
    score = evaluate(board)

    if score == 10 or score == -10:
        return score - depth if score == 10 else score + depth

    if is_draw(board):
        return 0

    if is_maximizing:
        max_eval = -math.inf
        for row in range(3):
            for col in range(3):
                if board[row][col] == EMPTY:
                    board[row][col] = PLAYER_X
                    eval = minimax(board, depth + 1, False, alpha, beta)
                    board[row][col] = EMPTY
                    max_eval = max(max_eval, eval)
                    alpha = max(alpha, eval)
                    if beta <= alpha:
                        break
        return max_eval

    else:
        min_eval = math.inf
        for row in range(3):
            for col in range(3):
                if board[row][col] == EMPTY:
                    board[row][col] = PLAYER_O
                    eval = minimax(board, depth + 1, True, alpha, beta)
                    board[row][col] = EMPTY
                    min_eval = min(min_eval, eval)
                    beta = min(beta, eval)
                    if beta <= alpha:
                        break
        return min_eval

def best_move(board):
    """Finds the best move for the AI."""
    best_val = -math.inf
    move = (-1, -1)
    for row in range(3):
        for col in range(3):
            if board[row][col] == EMPTY:
                board[row][col] = PLAYER_X
                move_val = minimax(board, 0, False, -math.inf, math.inf)
                board[row][col] = EMPTY

                if move_val > best_val:
                    best_val = move_val
                    move = (row, col)

    return move

def play():
    """Runs a Tic-Tac-Toe game between a human and AI."""
    board = [[EMPTY] * 3 for _ in range(3)]
    print("Tic-Tac-Toe: AI (X) vs. You (O)")
    print_board(board)

    while True:
        row, col = map(int, input("Enter your move (row and column 0-2): ").split())

        if board[row][col] != EMPTY:
            print("Invalid move! Try again.")
            continue

        board[row][col] = PLAYER_O

        if is_winner(board, PLAYER_O):
            print_board(board)
            print("You win! 🎉")
            break

        if is_draw(board):
            print_board(board)
            print("It's a draw! 🤝")
            break

        ai_row, ai_col = best_move(board)
        board[ai_row][ai_col] = PLAYER_X
        print("\nAI plays:")
        print_board(board)

        if is_winner(board, PLAYER_X):
            print("AI wins! 🤖")
            break

        if is_draw(board):
            print("It's a draw! 🤝")
            break



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

```
Tic-Tac-Toe: AI (X) vs. You (O)
  |   |
---------
  |   |
---------
  |   |
---------
Enter your move (row and column 0-2): 0 1

AI plays:
X | O |
---------
  |   |
---------
  |   |
---------
Enter your move (row and column 0-2): 1 1

AI plays:
X | O |
---------
  | O |
---------
  | X |
---------
Enter your move (row and column 0-2): 2 0

AI plays:
```

```
X | O | X
---------
  | O |
---------
O | X |
---------
Enter your move (row and column 0-2): 1 0

AI plays:
X | O | X
---------
O | O | X
---------
O | X |
---------
Enter your move (row and column 0-2): 2 2
X | O | X
---------
O | O | X
---------
O | X | O
---------
It's a draw! 🤝
```