```python
import math

# Board is a 3x3 list of lists
def create_board():
    return [[' ' for _ in range(3)] for _ in range(3)]

def print_board(board):
    for row in board:
        print('|'.join(row))
        print('-' * 5)

def is_winner(board, player):
    # Rows, columns and diagonals
    for i in range(3):
        if all(cell == player for cell in board[i]):  # Row
            return True
        if all(board[j][i] == player for j in range(3)):  # Column
            return True
    if all(board[i][i] == player for i in range(3)):  # Diagonal
        return True
    if all(board[i][2 - i] == player for i in range(3)):  # Anti-diagonal
        return True
    return False

def is_full(board):
    return all(cell != ' ' for row in board for cell in row)

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

def minimax(board, is_maximizing):
    if is_winner(board, 'O'):
        return 1
    if is_winner(board, 'X'):
        return -1
    if is_full(board):
        return 0

    if is_maximizing:
        best_score = -math.inf
        for (i, j) in get_available_moves(board):
            board[i][j] = 'O'
            score = minimax(board, False)
            board[i][j] = ' '
            best_score = max(score, best_score)
        return best_score
    else:
        best_score = math.inf
        for (i, j) in get_available_moves(board):
            board[i][j] = 'X'
            score = minimax(board, True)
            board[i][j] = ' '
            best_score = min(score, best_score)
        return best_score

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

def play_game():
    board = create_board()
    print("Welcome to Tic-Tac-Toe! You are X, AI is O.")
    print_board(board)

    while True:
        # Human move
        while True:
            try:
                x, y = map(int, input("Enter your move (row and column: 0-2): ").split())
                if board[x][y] == ' ':
                    board[x][y] = 'X'
                    break
                else:
                    print("Cell already taken.")
            except:
                print("Invalid input. Try again.")

        print_board(board)
        if is_winner(board, 'X'):
            print("You win!")
            break
        if is_full(board):
            print("It's a draw!")
            break

        # AI move
        ai_move = best_move(board)
        if ai_move:
            board[ai_move[0]][ai_move[1]] = 'O'
            print("AI played:")
            print_board(board)

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

if __name__ == "__main__":
    play_game()

```