In [1]:

```
import random

# Function to print the tic-tac-toe board
def print_board(board):
    for row in board:
        print(" | ".join(row))
        print("-" * 9)

# Function to check for a winning condition
def check_win(board, player):
    # Check rows
    for row in board:
        if all(mark == player for mark in row):
            return True
    # Check columns
    for col in range(3):
        if all(board[row][col] == player for row in range(3)):
            return True
    # Check diagonals
    if board[0][0] == board[1][1] == board[2][2] == player or board[0][2] == board[1
        return True
    return False

# Function to check for a tie condition
def check_tie(board):
    return all(board[row][col] != " " for row in range(3) for col in range(3))

# Function to make the AI move using the minimax algorithm
def make_ai_move(board, player):
    best_score = float("-inf")
    best_move = None

    for row in range(3):
        for col in range(3):
            if board[row][col] == " ":
                board[row][col] = player
                score = minimax(board, 0, False)
                board[row][col] = " "
                if score > best_score:
                    best_score = score
                    best_move = (row, col)

    board[best_move[0]][best_move[1]] = player

# Minimax algorithm
def minimax(board, depth, is_maximizing):
    scores = {
        "X": 1,
        "O": -1,
        "tie": 0
    }

    if check_win(board, "X"):
        return scores["X"]
    elif check_win(board, "O"):
        return scores["O"]
    elif check_tie(board):
        return scores["tie"]
```

```
    if is_maximizing:
        best_score = float("-inf")
        for row in range(3):
            for col in range(3):
                if board[row][col] == " ":
                    board[row][col] = "X"
                    score = minimax(board, depth + 1, False)
                    board[row][col] = " "
                    best_score = max(score, best_score)
        return best_score
    else:
        best_score = float("inf")
        for row in range(3):
            for col in range(3):
                if board[row][col] == " ":
                    board[row][col] = "O"
                    score = minimax(board, depth + 1, True)
                    board[row][col] = " "
                    best_score = min(score, best_score)
        return best_score

# Main game loop
def play_tic_tac_toe():
    board = [[" " for _ in range(3)] for _ in range(3)]
    players = ["X", "O"]
    current_player = random.choice(players)

    print("Welcome to Tic-Tac-Toe!")
    print_board(board)

    while True:
        print(f"It's Player {current_player}'s turn.")
        if current_player == "X":
            row = int(input("Enter the row (0-2): "))
            col = int(input("Enter the column (0-2): "))
            if board[row][            col] == " ":
                board[row][col] = current_player
            else:
                print("Invalid move. Try again.")
                continue
        else:
            make_ai_move(board, current_player)

        print_board(board)
        if check_win(board, current_player):
            print(f"Player {current_player} wins!")
            break
        elif check_tie(board):
            print("It's a tie!")
            break

        current_player = players[(players.index(current_player) + 1) % 2]

play_tic_tac_toe()
```

```
Welcome to Tic-Tac-Toe!
  |   |
---------
  |   |
---------
  |   |
---------
It's Player O's turn.
O |   |
---------
  |   |
---------
  |   |
---------
It's Player X's turn.
Enter the row (0-2): 0
Enter the column (0-2): 1
O | X |
---------
  |   |
---------
  |   |
---------
It's Player O's turn.
O | X | O
---------
  |   |
---------
  |   |
---------
It's Player X's turn.
Enter the row (0-2): 1
Enter the column (0-2): 1
O | X | O
---------
  | X |
---------
  |   |
---------
It's Player O's turn.
O | X | O
---------
O | X |
---------
  |   |
---------
It's Player X's turn.
Enter the row (0-2): 0
Enter the column (0-2): 1
Invalid move. Try again.
It's Player X's turn.
Enter the row (0-2): 0
Enter the column (0-2): 2
Invalid move. Try again.
It's Player X's turn.
Enter the row (0-2): 2
Enter the column (0-2): 0
O | X | O
---------
O | X |
---------
```

```
X |   |
---------
It's Player O's turn.
O | X | O
---------
O | X |
---------
X | O |
---------
It's Player X's turn.
Enter the row (0-2): 1
Enter the column (0-2): 2
O | X | O
---------
O | X | X
---------
X | O |
---------
It's Player O's turn.
O | X | O
---------
O | X | X
---------
X | O | O
---------
It's a tie!
```

In [ ]: