Họ tên: Phạm Văn Thuận

MSSV: 6351071068
Lớp: CQ.63.CNTT

Trò chơi Caro với Bot bằng Minimax

```
using System;
using System.Collections.Generic;


namespace CaroMinimax
{
   class Program
   {
      static char[,] board = {
         { ' ', ' ', ' ' },
         { ' ', ' ', ' ' },
         { ' ', ' ', ' ' }
      };


      static void Main(string[] args)
      {
         Console.OutputEncoding = System.Text.Encoding.UTF8;
         Console.WriteLine("=== CARO 3x3 – AI sử dụng thuật toán Minimax ===\n");


         while (true)
         {
            PrintBoard();
            Console.WriteLine("\nNgười chơi (X) – nhập vị trí (0-8):");
            int playerMove;
            while (!int.TryParse(Console.ReadLine(), out playerMove) || playerMove < 0 ||
playerMove > 8 || board[playerMove / 3, playerMove % 3] != ' ')
            {
                Console.Write("Không hợp lệ, nhập lại (0-8): ");
            }

            board[playerMove / 3, playerMove % 3] = 'X';

            if (CheckWinner() != ' ')
            {
                PrintBoard();
                Console.WriteLine($"\nNgười chơi {CheckWinner()} thắng!");
                break;
            }

            if (IsFull())
            {
                PrintBoard();
                Console.WriteLine("\nHòa!");
                break;
            }

            Console.WriteLine("\nAI (O) đang suy nghĩ...");
```

```
        (int bestScore, int bestMove) = Minimax(board, 0, true);

        int row = bestMove / 3;
        int col = bestMove % 3;
        board[row, col] = 'O';

        if (CheckWinner() != ' ')
        {
            PrintBoard();
            Console.WriteLine($"\nNgười chơi {CheckWinner()} thắng!");
            break;
        }

        if (IsFull())
        {
            PrintBoard();
            Console.WriteLine("\nHòa!");
            break;
        }
    }

    Console.WriteLine("\nKết thúc trò chơi.");
}

static void PrintBoard()
{
```

```
            Console.WriteLine("\n-------");
            for (int i = 0; i < 3; i++)
            {
                Console.Write("|");
                for (int j = 0; j < 3; j++)
                {
                    Console.Write(board[i, j]);
                    Console.Write("|");
                }
                Console.WriteLine();
                Console.WriteLine("-------");
            }
        }

        static char CheckWinner()
        {
            for (int i = 0; i < 3; i++)
            {
                if (board[i, 0] != ' ' && board[i, 0] == board[i, 1] && board[i, 1] == board[i,
2])
                    return board[i, 0];

                if (board[0, i] != ' ' && board[0, i] == board[1, i] && board[1, i] == board[2,
i])
                    return board[0, i];
            }
            if (board[0, 0] != ' ' && board[0, 0] == board[1, 1] && board[1, 1] == board[2,
2])
                return board[0, 0];

            if (board[0, 2] != ' ' && board[0, 2] == board[1, 1] && board[1, 1] == board[2,
0])
                return board[0, 2];

            return ' ';
        }

        static bool IsFull()
        {
            foreach (char cell in board)
            {
                if (cell == ' ') return false;
            }
            return true;
        }

        static (int score, int move) Minimax(char[,] currentBoard, int depth, bool
isMaximizing)
        {
            char winner = CheckWinner();
            if (winner == 'O') return (10 - depth, -1);
            if (winner == 'X') return (-10 + depth, -1);
            if (IsFull()) return (0, -1);

            int bestMove = -1;
            int bestScore = isMaximizing ? int.MinValue : int.MaxValue;

            for (int i = 0; i < 9; i++)
            {
                int row = i / 3;
                int col = i % 3;

                if (currentBoard[row, col] != ' ') continue;

                currentBoard[row, col] = isMaximizing ? 'O' : 'X';
                (int score, _) = Minimax(currentBoard, depth + 1, !isMaximizing);
                currentBoard[row, col] = ' ';

                Console.WriteLine($"[Depth {depth}] {(isMaximizing ? "MAX" : "MIN")} xét
ô {i} → điểm = {score}");

                if (isMaximizing)
                {
                    if (score > bestScore)
                    {
                        bestScore = score;
                        bestMove = i;
                    }
                }
                else
                {
                    if (score < bestScore)
                    {
                        bestScore = score;
                        bestMove = i;
                    }
                }
            }

            return (bestScore, bestMove);
        }
    }
}
```