**Name: Tavishi Mahajan**

**Class: TY-CS**

**Division: D**

**Roll no: - 56**

**Assignment No.: 1**

**Problem Statement: -** Implementation of AI and Non-AI techniques by implementing any two- players game.

**Technique Used: -** In implementation of AI and Non-AI techniques of any two player’s game. I have implemented tic-tac-toe using non-AI techniques. In this one the player is computer (local machine) and the other player is Human.

**Code: -**

import java.util.Random;

import java.util.Scanner;

public class tictactoe {

    private static final int[][] MAGIC\_SQUARE = {

            {2, 7, 6},

            {9, 5, 1},

            {4, 3, 8}

    };

    private static final char EMPTY = '-';

    private static final char PLAYER\_X = 'X';

    private static final char PLAYER\_O = 'O';

    private char[][] board;

    private char currentPlayer;

    public tictactoe() {

        board = new char[3][3];

        currentPlayer = PLAYER\_X;

        initializeBoard();

    }

    private void initializeBoard() {

        for (int i = 0; i < 3; i++) {

            for (int j = 0; j < 3; j++) {

                board[i][j] = EMPTY;

            }

        }

    }

    private void printBoard() {

        for (int i = 0; i < 3; i++) {

            for (int j = 0; j < 3; j++) {

                System.out.print(board[i][j] + " ");

            }

            System.out.println();

        }

    }

    private boolean isBoardFull() {

        for (int i = 0; i < 3; i++) {

            for (int j = 0; j < 3; j++) {

                if (board[i][j] == EMPTY) {

                    return false;

                }

            }

        }

        return true;

    }

    private boolean isMagicSquare() {

        int sum = 0;

        for (int i = 0; i < 3; i++) {

            for (int j = 0; j < 3; j++) {

                if (board[i][j] == currentPlayer) {

                    sum += MAGIC\_SQUARE[i][j];

                }

            }

        }

        for (int i = 0; i < 3; i++) {

            int rowSum = 0;

            int colSum = 0;

            for (int j = 0; j < 3; j++) {

                rowSum += MAGIC\_SQUARE[i][j];

                colSum += MAGIC\_SQUARE[j][i];

            }

            if (rowSum == sum || colSum == sum) {

                return true;

            }

        }

        int diagSum1 = MAGIC\_SQUARE[0][0] + MAGIC\_SQUARE[1][1] + MAGIC\_SQUARE[2][2];

        int diagSum2 = MAGIC\_SQUARE[0][2] + MAGIC\_SQUARE[1][1] + MAGIC\_SQUARE[2][0];

        return diagSum1 == sum || diagSum2 == sum;

    }

    private boolean makeMove(int row, int col) {

        if (row < 0 || row >= 3 || col < 0 || col >= 3 || board[row][col] != EMPTY) {

            return false;

        }

        board[row][col] = currentPlayer;

        return true;

    }

    private void switchPlayer() {

        currentPlayer = (currentPlayer == PLAYER\_X) ? PLAYER\_O : PLAYER\_X;

    }

    public void play() {

    Scanner scanner = new Scanner(System.in);

    Random random = new Random();

    while (true) {

        System.out.println("Current board:");

        printBoard();

        if (currentPlayer == PLAYER\_X) {

            System.out.println("Player " + currentPlayer + ", make your move (row and column):");

            int row = scanner.nextInt();

            int col = scanner.nextInt();

            if (!makeMove(row, col)) {

                System.out.println("Invalid move! Try again.");

                continue;

            }

            if (isMagicSquare()) {

            System.out.println("Player " + currentPlayer + " wins!");

            break;

        }

        } else {

            System.out.println("Computer (Player " + currentPlayer + ") is making a move...");

            int row, col;

            do {

                row = random.nextInt(3);

                col = random.nextInt(3);

                if (isMagicSquare()) {

                System.out.println("Player " + currentPlayer + " wins!");

                break;

            }

            } while (!makeMove(row, col));

        }

        if (isMagicSquare()) {

            System.out.println("Player " + currentPlayer + " wins!");

            break;

        }

        if (isBoardFull()) {

            System.out.println("It's a draw!");

            break;

        }

        switchPlayer();

    }

}

    public static void main(String[] args) {

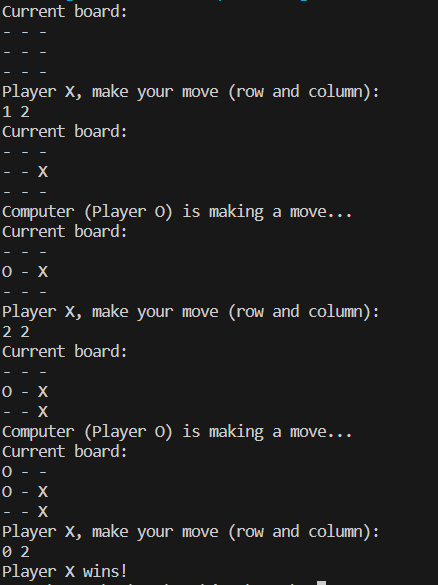
        tictactoe ticTacToe = new tictactoe();

        ticTacToe.play();

    }

}

**Output: -**

****