Untitled-2

```java
import java.util.Scanner;

public class TicTacToe {
    private static final int SIZE = 3;
    private static char[][] board = new char[SIZE][SIZE];
    private static char currentPlayer = 'X';

    public static void main(String[] args) {
        initializeBoard();
        Scanner scanner = new Scanner(System.in);
        boolean gameWon = false;
        boolean gameDraw = false;

        while (!gameWon && !gameDraw) {
            printBoard();
            playerMove(scanner);
            gameWon = checkForWin();
            gameDraw = checkForDraw();
            if (!gameWon && !gameDraw) {
                switchPlayer();
            }
        }

        printBoard();
        if (gameWon) {
            System.out.println("Player " + currentPlayer + " wins!");
        } else {
            System.out.println("The game is a draw!");
        }

        scanner.close();
    }

    private static void initializeBoard() {
        for (int i = 0; i < SIZE; i++) {
            for (int j = 0; j < SIZE; j++) {
                board[i][j] = ' ';
            }
        }
    }

    private static void printBoard() {
        System.out.println();
        for (int i = 0; i < SIZE; i++) {
            for (int j = 0; j < SIZE; j++) {
                System.out.print(board[i][j]);
                if (j < SIZE - 1) System.out.print(" | ");
            }
            System.out.println();
            if (i < SIZE - 1) {
                System.out.println("--|---|--");
            }
        }
        System.out.println();
    }

    private static void playerMove(Scanner scanner) {
        int row, col;
        while (true) {
            System.out.println("Player " + currentPlayer + ", enter your move (row and column): ");
            row = scanner.nextInt() - 1;
            col = scanner.nextInt() - 1;

            if (row >= 0 && row < SIZE && col >= 0 && col < SIZE && board[row][col] == ' ') {
                board[row][col] = currentPlayer;
                break;
            } else {
                System.out.println("This move is not valid. Try again.");
            }
        }
    }

    private static void switchPlayer() {
        currentPlayer = (currentPlayer == 'X') ? 'O' : 'X';
    }

    private static boolean checkForWin() {
        // Check rows and columns
        for (int i = 0; i < SIZE; i++) {
            if (checkRow(i) || checkColumn(i)) {
                return true;
            }
        }
        // Check diagonals
        return checkMainDiagonal() || checkAntiDiagonal();
    }

    private static boolean checkRow(int row) {
        char symbol = board[row][0];
        return symbol != ' ' && symbol == board[row][1] && symbol == board[row][2];
    }

    private static boolean checkColumn(int col) {
        char symbol = board[0][col];
        return symbol != ' ' && symbol == board[1][col] && symbol == board[2][col];
    }

    private static boolean checkMainDiagonal() {
        char symbol = board[0][0];
        return symbol != ' ' && symbol == board[1][1] && symbol == board[2][2];
    }

    private static boolean checkAntiDiagonal() {
        char symbol = board[0][2];
        return symbol != ' ' && symbol == board[1][1] && symbol == board[2][0];
    }

    private static boolean checkForDraw() {
        for (int i = 0; i < SIZE; i++) {
            for (int j = 0; j < SIZE; j++) {
                if (board[i][j] == ' ') {
                    return false; // There is at least one empty space
                }
            }
        }
        return true; // No empty spaces left
    }
}
```