

# TIC TAC TOE GAME PROJECT

```
/*
```

```
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change  
this license
```

```
*/
```

```
package com.mycompany.tictactoe;
```

```
import java.util.Scanner;
```

```
public class TicTacToe {
```

```
    static Scanner input = new Scanner(System.in);
```

```
    static char[][] board = new char[3][3];
```

```
    static char currentPlayer = 'X';
```

```
    public static void main(String[] args) {
```

```
        initializeBoard();
```

```
        printBoard();
```

```
        while (!gameOver()) {
```

```
            makeMove();
```

```
            printBoard();
```

```
            switchPlayer();
```

```
        }
```

```
        System.out.println("Game Over!");
```

```
    }
```

```
    public static void initializeBoard() {
```

```
        for (int i = 0; i < 3; i++) {
```

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

```
                board[i][j] = '-';
```

```
    }  
    }  
}
```

```
public static void printBoard() {  
    System.out.println("-----");  
    for (int i = 0; i < 3; i++) {  
        System.out.print("| ");  
        for (int j = 0; j < 3; j++) {  
            System.out.print(board[i][j] + " | ");  
        }  
        System.out.println();  
        System.out.println("-----");  
    }  
}
```

```
public static void makeMove() {  
    System.out.println("Player " + currentPlayer + ", enter row (1-3): ");  
    int row = input.nextInt() - 1;  
    System.out.println("Player " + currentPlayer + ", enter column (1-3): ");  
    int col = input.nextInt() - 1;  
    board[row][col] = currentPlayer;  
}
```

```
public static boolean gameOver() {  
    if (checkRows() || checkColumns() || checkDiagonals()) {  
        System.out.println("Player " + currentPlayer + " wins!");  
        return true;  
    } else if (boardFull()) {  
        System.out.println("Tie game!");  
    }  
}
```

```
        return true;
    } else {
        return false;
    }
}
```

```
public static boolean checkRows() {
    for (int i = 0; i < 3; i++) {
        if (board[i][0] == board[i][1] && board[i][1] == board[i][2] && board[i][0] != '-') {
            return true;
        }
    }
    return false;
}
```

```
public static boolean checkColumns() {
    for (int i = 0; i < 3; i++) {
        if (board[0][i] == board[1][i] && board[1][i] == board[2][i] && board[0][i] != '-') {
            return true;
        }
    }
    return false;
}
```

```
public static boolean checkDiagonals() {
    if (board[0][0] == board[1][1] && board[1][1] == board[2][2] && board[0][0] != '-') {
        return true;
    } else if (board[0][2] == board[1][1] && board[1][1] == board[2][0] && board[0][2] !=
'-') {
        return true;
    } else {
```

```

        return false;
    }
}

public static boolean boardFull() {
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            if (board[i][j] == '-') {
                return false;
            }
        }
    }
    return true;
}

```

```

public static void switchPlayer() {
    if (currentPlayer == 'X') {
        currentPlayer = 'O';
    } else {
        currentPlayer = 'X';
    }
}
}

```

Output

```
-----  
| 0 | X | 0 |  
-----
```

```
| - | X | - |  
-----
```

```
| - | - | - |  
-----
```

Player X, enter row (1-3):

3

Player X, enter column (1-3):

2

```
-----  
| 0 | X | 0 |  
-----
```

```
| - | X | - |  
-----
```

```
| - | X | - |  
-----
```

Player 0 wins!

Game Over!

=== Code Execution Successful ===

```
-----
| 0 | X | - |
-----
| - | - | - |
-----
| - | - | - |
-----
Player X, enter row (1-3):
2
Player X, enter column (1-3):
2
-----
| 0 | X | - |
-----
| - | X | - |
-----
| - | - | - |
-----
Player 0, enter row (1-3):
1
Player 0, enter column (1-3):
3
```

```
-----
| - | - | - |
-----
| - | - | - |
-----
| - | - | - |
-----
Player X, enter row (1-3):
1
Player X, enter column (1-3):
2
-----
| - | X | - |
-----
| - | - | - |
-----
| - | - | - |
-----
Player O, enter row (1-3):
1
Player O, enter column (1-3):
1
```