```
import java.util.Scanner;

public class TicTacToe {
    static char[][] board = new char[3][3];
    static char currentPlayer = 'X';
    static boolean gameOver = false;

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

        while (!gameOver) {
            printBoard();
            playerMove(scanner);
            checkGameOver();
            switchPlayer();
        }

        printBoard();
        if (gameOver) {
            System.out.println("Game Over!");
        }
    }

    // Initialize the game board with empty spaces
    public static void initializeBoard() {
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                board[i][j] = ' ';
            }
        }
    }

    // Print the current board
    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("-------------");
        }
```

```
    }

    // Handle the player's move
    public static void playerMove(Scanner scanner) {
        int row, col;

        while (true) {
            System.out.println("Player " + currentPlayer + ", enter your move (row and column 0-2):
");
            row = scanner.nextInt();
            col = scanner.nextInt();

            if (row >= 0 && row < 3 && col >= 0 && col < 3 && board[row][col] == ' ') {
                board[row][col] = currentPlayer;
                break;
            } else {
                System.out.println("This move is not valid. Try again.");
            }
        }
    }

    // Check if the game is over (win or draw)
    public static void checkGameOver() {
        if (checkWin()) {
            gameOver = true;
            System.out.println("Player " + currentPlayer + " wins!");
        } else if (isBoardFull()) {
            gameOver = true;
            System.out.println("It's a draw!");
        }
    }

    // Check if any player has won
    public static boolean checkWin() {
        // Check rows, columns, and diagonals
        for (int i = 0; i < 3; i++) {
            if (board[i][0] == currentPlayer && board[i][1] == currentPlayer && board[i][2] ==
currentPlayer) {
                return true;
            }
            if (board[0][i] == currentPlayer && board[1][i] == currentPlayer && board[2][i] ==
currentPlayer) {
                return true;
            }
```

```
        }
        if (board[0][0] == currentPlayer && board[1][1] == currentPlayer && board[2][2] ==
currentPlayer) {
            return true;
        }
        if (board[0][2] == currentPlayer && board[1][1] == currentPlayer && board[2][0] ==
currentPlayer) {
            return true;
        }
        return false;
    }

    // Check if the board is full (i.e., the game is a draw)
    public static boolean isBoardFull() {
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                if (board[i][j] == ' ') {
                    return false;
                }
            }
        }
        return true;
    }

    // Switch the current player between X and O
    public static void switchPlayer() {
        currentPlayer = (currentPlayer == 'X') ? 'O' : 'X';
    }
}
```