```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class TicTacToeGUI {
    private static char[][] board = {{' ', ' ', ' '}, {' ', ' ', ' '}, {' ', ' ', ' '}};
    private static char currentPlayer = 'X';
    private static boolean gameWon = false;
    private static JButton[][] buttons = new JButton[3][3];
    private static JLabel statusLabel;

    public static void main(String[] args) {
        // Create the main frame
        JFrame frame = new JFrame("Tic-Tac-Toe");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(400, 450);
        frame.setLayout(new BorderLayout());

        // Create the status label
        statusLabel = new JLabel("Player X's turn", JLabel.CENTER);
        statusLabel.setFont(new Font("Arial", Font.PLAIN, 20));
        frame.add(statusLabel, BorderLayout.NORTH);

        // Create the game grid
        JPanel gridPanel = new JPanel();
        gridPanel.setLayout(new GridLayout(3, 3));
        frame.add(gridPanel, BorderLayout.CENTER);

        // Initialize buttons for each cell
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                buttons[i][j] = new JButton(" ");
                buttons[i][j].setFont(new Font("Arial", Font.PLAIN, 60));
                buttons[i][j].setFocusPainted(false);
                buttons[i][j].setEnabled(true);
                buttons[i][j].addActionListener(new ButtonClickListener(i, j));
                gridPanel.add(buttons[i][j]);
            }
        }

        frame.setVisible(true);
    }

    // ActionListener for each button click
    static class ButtonClickListener implements ActionListener {
        private int row, col;

        public ButtonClickListener(int row, int col) {
            this.row = row;
            this.col = col;
        }

        @Override
        public void actionPerformed(ActionEvent e) {
            if (board[row][col] == ' ' && !gameWon) {
```

```
board[row][col] = currentPlayer;
buttons[row][col].setText(String.valueOf(currentPlayer));
buttons[row][col].setEnabled(false);

if (checkWin()) {
    gameWon = true;
    statusLabel.setText("Player " + currentPlayer + " wins!");
} else {
    switchPlayer();
    statusLabel.setText("Player " + currentPlayer + "'s turn");
}
            }
        }
    }

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

    // Check for a winning condition
    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] == currentPlaye
r) ||
                (board[0][i] == currentPlayer && board[1][i] == currentPlayer && board[2][i] == currentPlaye
r)) {
                return true;
            }
        }
        if ((board[0][0] == currentPlayer && board[1][1] == currentPlayer && board[2][2] == currentPlaye
r) ||
            (board[0][2] == currentPlayer && board[1][1] == currentPlayer && board[2][0] == currentPlaye
r)) {
            return true;
        }
        return false;
    }
}
```