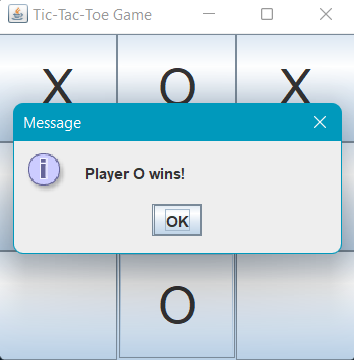
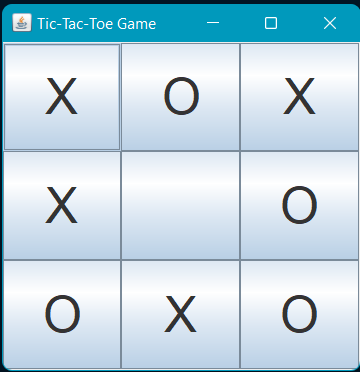
**Major Project: Tic Tac Toe Game**

import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.*ActionListener*;  
  
public class TicTacToeGame extends JFrame {  
  
 private JButton[][] buttons;  
 private char currentPlayer;  
  
 public TicTacToeGame() {  
 *// Set up the JFrame* super("Tic-Tac-Toe Game");  
 setSize(300, 300);  
 setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  
 setLocationRelativeTo(null);  
  
 *// Set up the components* buttons = new JButton[3][3];  
 currentPlayer = 'X';  
  
 *// Set up the layout* setLayout(new GridLayout(3, 3));  
  
 *// Create buttons and add action listener* for (int row = 0; row < 3; row++) {  
 for (int col = 0; col < 3; col++) {  
 buttons[row][col] = new JButton("");  
 buttons[row][col].setFont(new Font("Arial", Font.PLAIN, 40));  
 buttons[row][col].setFocusPainted(false);  
 buttons[row][col].addActionListener(new ButtonClickListener(row, col));  
 add(buttons[row][col]);  
 }  
 }  
 }  
  
 private 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 (buttons[row][col].getText().equals("")) {  
 buttons[row][col].setText(String.*valueOf*(currentPlayer));  
 if (checkWin()) {  
 JOptionPane.*showMessageDialog*(TicTacToeGame.this, "Player " + currentPlayer + " wins!");  
 resetGame();  
 } else if (checkDraw()) {  
 JOptionPane.*showMessageDialog*(TicTacToeGame.this, "It's a draw!");  
 resetGame();  
 } else {  
 switchPlayer();  
 }  
 } else {  
 JOptionPane.*showMessageDialog*(TicTacToeGame.this, "Cell already taken. Choose another cell.");  
 }  
 }  
 }  
  
 private void switchPlayer() {  
 currentPlayer = (currentPlayer == 'X') ? 'O' : 'X';  
 }  
  
 private boolean checkWin() {  
 *// Check rows, columns, and diagonals for a win* for (int i = 0; i < 3; i++) {  
 if (buttons[i][0].getText().equals(String.*valueOf*(currentPlayer)) &&  
 buttons[i][1].getText().equals(String.*valueOf*(currentPlayer)) &&  
 buttons[i][2].getText().equals(String.*valueOf*(currentPlayer))) {  
 return true; *// Row win* }  
 if (buttons[0][i].getText().equals(String.*valueOf*(currentPlayer)) &&  
 buttons[1][i].getText().equals(String.*valueOf*(currentPlayer)) &&  
 buttons[2][i].getText().equals(String.*valueOf*(currentPlayer))) {  
 return true; *// Column win* }  
 }  
 if (buttons[0][0].getText().equals(String.*valueOf*(currentPlayer)) &&  
 buttons[1][1].getText().equals(String.*valueOf*(currentPlayer)) &&  
 buttons[2][2].getText().equals(String.*valueOf*(currentPlayer))) {  
 return true; *// Diagonal win (top-left to bottom-right)* }  
 if (buttons[0][2].getText().equals(String.*valueOf*(currentPlayer)) &&  
 buttons[1][1].getText().equals(String.*valueOf*(currentPlayer)) &&  
 buttons[2][0].getText().equals(String.*valueOf*(currentPlayer))) {  
 return true; *// Diagonal win (top-right to bottom-left)* }  
 return false;  
 }  
  
 private boolean checkDraw() {  
 *// Check if all cells are filled* for (int i = 0; i < 3; i++) {  
 for (int j = 0; j < 3; j++) {  
 if (buttons[i][j].getText().equals("")) {  
 return false; *// Not a draw* }  
 }  
 }  
 return true; *// Draw* }  
  
 private void resetGame() {  
 *// Clear all cells and reset the current player* for (int i = 0; i < 3; i++) {  
 for (int j = 0; j < 3; j++) {  
 buttons[i][j].setText("");  
 }  
 }  
 currentPlayer = 'X';  
 }  
  
 public static void main(String[] args) {  
 SwingUtilities.*invokeLater*(new Runnable() {  
 @Override  
 public void run() {  
 new TicTacToeGame().setVisible(true);  
 }  
 });  
 }  
}

****