GURU NANAK COLLEGE BUDHLADA

DEPARTMENT: COMPUTER

NAME OF PROJECT: Tic Tac Toe

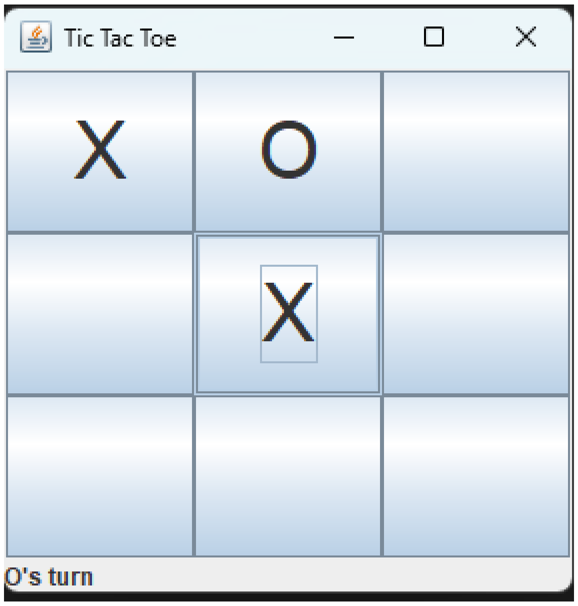
Submitted to:	Submitted by:
HOD	**Naveeta (321813)**
Dr. Rekha Kalra	**Happy Singh(321847)**
	Sameena(321856)

Table of Contents

1. Introduction

The Tic Tac Toe Game is a classic two-player game where players take turns in a 3x3 grid with symbols, "X" and "O". using Java programming language with a graphical user interface (GUI).

Preview: 1

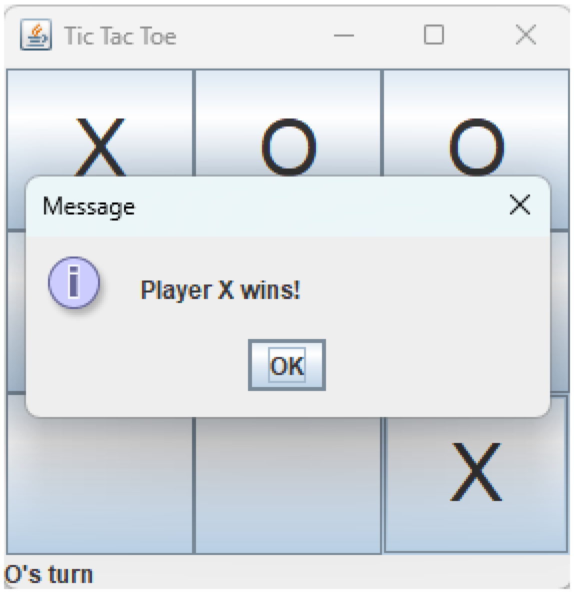


2. Functionality

The implemented Tic Tac Toe game provides the following functionalities:

- GUIe: The game is played using a graphical user interface
- Two-Player Mode: two players can play one use X and another O
- Win: The game automatically detects when a player wins by getting three of their symbols in a row, column, or diagonal.
- Draw Detection: The game detects when there are no more empty spaces and declares a draw if no player wins.

Preview 2:



Message

Player X wins!

O's turn

3. Code Explanation

The code is structured into a single Java class, TicTacToeGUI, which extends the JFrame class to create the GUI window.

- Action Listeners: Action listeners are used to handle button clicks and update the game state accordingly.
- Win and Draw Detection: Methods are implemented to check for win and draw conditions after each move.
- Game Reset: The game can be reset after a win or draw to start a new game.
- GUI Setup: The GUI layout consists of a 3x3 grid.

CODE

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



public class TicTacToeGUI extends JFrame implements ActionListener {
    private JButton[][] buttons;
    private JLabel statusLabel;

   private int currentPlayer;



    public TicTacToeGUI() {
        setTitle("Tic Tac Toe");
        setSize(300, 300);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());

        JPanel boardPanel = new JPanel();
        boardPanel.setLayout(new GridLayout(3, 3));
        buttons = new JButton[3][3];
        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, 40));
                buttons[i][j].addActionListener(this);
                boardPanel.add(buttons[i][j]);
            }
        }

        add(boardPanel, BorderLayout.CENTER);

        statusLabel = new JLabel("X's turn");
        add(statusLabel, BorderLayout.SOUTH);



        currentPlayer = 1; // Player 1 starts the game
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        JButton buttonClicked = (JButton) e.getSource();


        if (buttonClicked.getText().equals("")) {
            if (currentPlayer == 1) {
                buttonClicked.setText("X");
                currentPlayer = 2;
                statusLabel.setText("O's turn");
            } else {
                buttonClicked.setText("O");
                currentPlayer = 1;
                statusLabel.setText("X's turn");
            }

            if (checkForWin()) {
                JOptionPane.showMessageDialog(this, "Player " + (currentPlayer == 1 ? "O" : "X") + "
wins!");
                resetGame();
            } else if (checkForDraw()) {
                JOptionPane.showMessageDialog(this, "It's a draw!");
                resetGame();
            }
        }
    }
    private boolean checkForWin() {
        String[][] board = new String[3][3];
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                board[i][j] = buttons[i][j].getText();
            }
        }
        // Check rows
        for (int i = 0; i < 3; i++) {
            if (board[i][0].equals(board[i][1]) && board[i][0].equals(board[i][2]) &&
!board[i][0].equals("")) {
                return true;
            }
        }
        // Check columns
        for (int j = 0; j < 3; j++) {
            if (board[0][j].equals(board[1][j]) && board[0][j].equals(board[2][j]) &&
!board[0][j].equals("")) {
                return true;
            }
        }
        // Check diagonals
        if (board[0][0].equals(board[1][1]) && board[0][0].equals(board[2][2]) &&
!board[0][0].equals("")) {
            return true;
        }
        if (board[0][2].equals(board[1][1]) && board[0][2].equals(board[2][0]) &&
!board[0][2].equals("")) {
            return true;
        }
        return false;
    }

    private boolean checkForDraw() {
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                if (buttons[i][j].getText().equals("")) {
                    return false;
                }
            }
        }
        return true;
    }
    private void resetGame() {
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                buttons[i][j].setText("");
            }
        }
        currentPlayer = 1;
        statusLabel.setText("X's turn");
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            TicTacToeGUI game = new TicTacToeGUI();
            game.setVisible(true);
        });
    }
}
```

4. Guide and Rules

- Player has to complete a row column or diagonal to win
- After win the winner symbol will be shown
- Dont let other player to complete 3 Os or Xs frist in line
- Game declares draw if no player is able to complete a line or 3.

6 . Conclusion

In conclusion, this project successfully implements a simple Tic Tac Toe game using Java with a graphical user interface. It provides an interactive and enjoyable gaming experience for two players.