

CodeTechSolution

Task -1 :-

```
package com.codetech.TicTacToeGUI;

import javax.swing.*;

import java.awt.*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.*;

public class TicTacToeGUI extends JFrame{

    private char[][] board = {{' ', ' ', ' '}, {' ', ' ', ' '}, {' ', ' ', ' '}};

    private char currentPlayer = 'X';

    private JButton[][] buttons = new JButton[3][3];

    public TicTacToeGUI() {

        setTitle("Tic-Tac-Toe");

        setSize(500, 500);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        setLocationRelativeTo(null);

        setLayout(new GridLayout(3, 3));

        initializeButtons();

        initializeMenu();

        setVisible(true);

    }

    private void initializeButtons() {

        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(Font.SANS_SERIF, Font.BOLD, 70));

buttons[i][j].setFocusPainted(false);

final int row = i;

final int col = j;

buttons[i][j].addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

if (board[row][col] == ' ' && currentPlayer != ' ') {

buttons[row][col].setText(String.valueOf(currentPlayer));

board[row][col] = currentPlayer;

if (isWinner()) {

JOptionPane.showMessageDialog(null, "Player " + currentPlayer + " wins!");

resetGame();

} else if (isBoardFull()) {

JOptionPane.showMessageDialog(null, "Oop's The game is a tie!");

resetGame();

} else {

switchPlayer();

}

}

});

add(buttons[i][j]);

}

}

}

private void initializeMenu() {

JMenuBar menuBar = new JMenuBar();

```

```
JMenu fileMenu = new JMenu("File");

JMenuItem saveItem = new JMenuItem("Save");

JMenuItem loadItem = new JMenuItem("Load");

JMenuItem exitItem = new JMenuItem("Exit");

saveItem.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

saveGame();

}

});

loadItem.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

loadGame();

}

});

exitItem.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

System.exit(0);

}

});

fileMenu.add(saveItem);

fileMenu.add(loadItem);

fileMenu.add(exitItem);

menuBar.add(fileMenu);

setJMenuBar(menuBar);

}
```

```
private void resetGame() {

    currentPlayer = 'X';

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

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

            buttons[i][j].setText("");

            board[i][j] = ' ';

        }

    }

}

private void switchPlayer() {

    currentPlayer = (currentPlayer == 'X') ? 'O' : 'X';

}

private boolean isWinner() {

    // Check rows

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

        if (board[i][0] == currentPlayer && board[i][1] == currentPlayer &&
            board[i][2] == currentPlayer) {

            return true;

        }

    }

    // Check columns

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

        if (board[0][i] == currentPlayer && board[1][i] == currentPlayer &&
            board[2][i] == currentPlayer) {

            return true;

        }

    }

    // Check diagonals
```

```

if ((board[0][0] == currentPlayer && board[1][1] == currentPlayer &&
board[2][2] == currentPlayer) ||

(board[0][2] == currentPlayer && board[1][1] == currentPlayer &&
board[2][0] == currentPlayer)) {

    return true;

}

return false;

}

private boolean isBoardFull() {

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

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

            if (board[i][j] == ' ') {

                return false; // If there is an empty cell, the board is not full

            }

        }

    }

    return true;

}

private void saveGame() {

    try (ObjectOutputStream out = new ObjectOutputStream(new
        FileOutputStream("tictactoe.dat"))) {

        out.writeObject(board);

        out.writeChar(currentPlayer);

        JOptionPane.showMessageDialog(null, "Game saved successfully!");

    } catch (IOException e) {

        e.printStackTrace();

    }

}

private void loadGame() {

```

```

try (ObjectInputStream in = new ObjectInputStream(new
FileInputStream("tictactoe.dat"))) {

    board = (char[][] in.readObject());

    currentPlayer = in.readChar();

    updateButtons();

    JOptionPane.showMessageDialog(null, "Game loaded successfully!");

} catch (IOException | ClassNotFoundException e) {

    e.printStackTrace();

}

}

private void updateButtons() {

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

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

            buttons[i][j].setText(String.valueOf(board[i][j]));

        }

    }

}

public static void main(String[] args) {

    SwingUtilities.invokeLater(new Runnable() {

        @Override

        public void run() {

            new TicTacToeGUI();

        }

    });

}

}

```

