**SMART TRAFFIC MANAGEMENT**

import javax.swing.\*;

import java.awt.\*;

public class TrafficLightAWT extends JFrame {

private JPanel lightPanel;

private JLabel redLight1, yellowLight1, greenLight1;

private JLabel redLight2, yellowLight2, greenLight2;

private JLabel redLight3, yellowLight3, greenLight3;

private static final Color RED\_LIGHT = Color.RED;

private static final Color YELLOW\_LIGHT = Color.YELLOW;

private static final Color GREEN\_LIGHT = Color.GREEN;

private static final Color OFF\_LIGHT = Color.GRAY;

private int[] vehicleCounts; // Stores the number of vehicles for each direction

public TrafficLightAWT() {

setTitle("Smart Traffic Light with Vehicle Input");

setSize(600, 600);

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

setLocationRelativeTo(null);

setLayout(new BorderLayout());

// Input Panel

JPanel inputPanel = new JPanel(new GridLayout(4, 2, 10, 10));

inputPanel.setBorder(BorderFactory.createTitledBorder("Enter Vehicle Counts"));

JLabel road1Label = new JLabel("Road 1 (Top):");

JLabel road2Label = new JLabel("Road 2 (Middle):");

JLabel road3Label = new JLabel("Road 3 (Bottom):");

JTextField road1Field = new JTextField();

JTextField road2Field = new JTextField();

JTextField road3Field = new JTextField();

JButton submitButton = new JButton("Submit");

inputPanel.add(road1Label);

inputPanel.add(road1Field);

inputPanel.add(road2Label);

inputPanel.add(road2Field);

inputPanel.add(road3Label);

inputPanel.add(road3Field);

inputPanel.add(new JLabel()); // Spacer

inputPanel.add(submitButton);

add(inputPanel, BorderLayout.NORTH);

// Traffic Light Panel for all roads

lightPanel = new JPanel();

lightPanel.setLayout(new GridLayout(3, 1));

// Road 1 Panel

JPanel road1Panel = createRoadPanel("Road 1 (Top)");

redLight1 = createTrafficLightLabel(OFF\_LIGHT);

yellowLight1 = createTrafficLightLabel(OFF\_LIGHT);

greenLight1 = createTrafficLightLabel(OFF\_LIGHT);

addLightsToPanel(road1Panel, redLight1, yellowLight1, greenLight1);

// Road 2 Panel

JPanel road2Panel = createRoadPanel("Road 2 (Middle)");

redLight2 = createTrafficLightLabel(OFF\_LIGHT);

yellowLight2 = createTrafficLightLabel(OFF\_LIGHT);

greenLight2 = createTrafficLightLabel(OFF\_LIGHT);

addLightsToPanel(road2Panel, redLight2, yellowLight2, greenLight2);

// Road 3 Panel

JPanel road3Panel = createRoadPanel("Road 3 (Bottom)");

redLight3 = createTrafficLightLabel(OFF\_LIGHT);

yellowLight3 = createTrafficLightLabel(OFF\_LIGHT);

greenLight3 = createTrafficLightLabel(OFF\_LIGHT);

addLightsToPanel(road3Panel, redLight3, yellowLight3, greenLight3);

lightPanel.add(road1Panel);

lightPanel.add(road2Panel);

lightPanel.add(road3Panel);

add(lightPanel, BorderLayout.CENTER);

// Button Action

submitButton.addActionListener(e -> {

try {

int road1Vehicles = Integer.parseInt(road1Field.getText());

int road2Vehicles = Integer.parseInt(road2Field.getText());

int road3Vehicles = Integer.parseInt(road3Field.getText());

vehicleCounts = new int[]{road1Vehicles, road2Vehicles, road3Vehicles};

// Update the traffic lights based on the vehicle count

updateTrafficLights();

} catch (NumberFormatException ex) {

JOptionPane.showMessageDialog(this, "Please enter valid numbers for all roads.", "Input Error", JOptionPane.ERROR\_MESSAGE);

}

});

}

private JPanel createRoadPanel(String title) {

JPanel panel = new JPanel(new FlowLayout());

panel.setBorder(BorderFactory.createTitledBorder(title));

return panel;

}

private JLabel createTrafficLightLabel(Color color) {

JLabel label = new JLabel();

label.setOpaque(true);

label.setBackground(color);

label.setPreferredSize(new Dimension(100, 100));

return label;

}

private void addLightsToPanel(JPanel panel, JLabel red, JLabel yellow, JLabel green) {

panel.add(red);

panel.add(yellow);

panel.add(green);

}

private void updateTrafficLights() {

// Determine the highest, middle, and lowest vehicle counts

int maxIndex = 0, midIndex = 0, minIndex = 0;

for (int i = 1; i < vehicleCounts.length; i++) {

if (vehicleCounts[i] > vehicleCounts[maxIndex]) {

maxIndex = i;

}

}

if ((maxIndex + 1) % 3 != maxIndex && vehicleCounts[(maxIndex + 1) % 3] >= vehicleCounts[(maxIndex + 2) % 3]) {

midIndex = (maxIndex + 1) % 3;

minIndex = (maxIndex + 2) % 3;

} else {

midIndex = (maxIndex + 2) % 3;

minIndex = (maxIndex + 1) % 3;

}

// Update lights

setLightColors(maxIndex, RED\_LIGHT, OFF\_LIGHT, OFF\_LIGHT);

setLightColors(midIndex, OFF\_LIGHT, YELLOW\_LIGHT, OFF\_LIGHT);

setLightColors(minIndex, OFF\_LIGHT, OFF\_LIGHT, GREEN\_LIGHT);

}

private void setLightColors(int road, Color red, Color yellow, Color green) {

switch (road) {

case 0:

redLight1.setBackground(red);

yellowLight1.setBackground(yellow);

greenLight1.setBackground(green);

break;

case 1:

redLight2.setBackground(red);

yellowLight2.setBackground(yellow);

greenLight2.setBackground(green);

break;

case 2:

redLight3.setBackground(red);

yellowLight3.setBackground(yellow);

greenLight3.setBackground(green);

break;

}

}

public static void main(String[] args) {

SwingUtilities.invokeLater(() -> {

TrafficLightAWT trafficLight = new TrafficLightAWT();

trafficLight.setVisible(true);

});

}

}