package chatappletsimulation;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.Random;

public class ChatAppletSimulation extends JFrame {

// Text area to display the chat messages

private JTextArea chatArea;

// Buttons for controlling the simulation

private JButton startButton, stopButton, pauseButton, resumeButton;

// User threads for simulating chat messages

private ChatUser user1, user2;

// Control variables to manage the chat flow

private volatile boolean isRunning = false;

private volatile boolean isPaused = false;

public ChatAppletSimulation() {

// Set up the JFrame layout

setTitle("Chat Simulation");

setSize(450, 400);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(new BorderLayout());

// Initialize the chat area

chatArea = new JTextArea();

chatArea.setEditable(false);

add(new JScrollPane(chatArea), BorderLayout.CENTER);

// Initialize control buttons

startButton = new JButton("Start");

stopButton = new JButton("Stop");

pauseButton = new JButton("Pause");

resumeButton = new JButton("Resume");

// Panel to hold the buttons

JPanel buttonPanel = new JPanel();

buttonPanel.add(startButton);

buttonPanel.add(stopButton);

buttonPanel.add(pauseButton);

buttonPanel.add(resumeButton);

add(buttonPanel, BorderLayout.SOUTH);

// Add button action listeners

startButton.addActionListener(new StartAction());

stopButton.addActionListener(new StopAction());

pauseButton.addActionListener(new PauseAction());

resumeButton.addActionListener(new ResumeAction());

}

// Inner class representing a chat user

class ChatUser extends Thread {

private String userName;

private Random random = new Random();

public ChatUser(String name) {

this.userName = name;

}

public void run() {

while (isRunning) {

// Check if the simulation is paused

if (!isPaused) {

// Simulate sending a message with random intervals

SwingUtilities.invokeLater(() -> chatArea.append(userName + ": Hello!\n"));

try {

// Wait for a random time between messages (1-3 seconds)

Thread.sleep(1000 + random.nextInt(2000));

} catch (InterruptedException e) {

Thread.currentThread().interrupt();

}

} else {

// If paused, yield the thread

Thread.yield();

}

}

}

}

// Action to start the chat simulation

private class StartAction implements ActionListener {

public void actionPerformed(ActionEvent e) {

if (!isRunning) {

isRunning = true;

isPaused = false;

chatArea.setText(""); // Clear the chat area

user1 = new ChatUser("User1");

user2 = new ChatUser("User2");

user1.start();

user2.start();

}

}

}

// Action to stop the chat simulation

private class StopAction implements ActionListener {

public void actionPerformed(ActionEvent e) {

isRunning = false;

isPaused = false;

chatArea.append("Chat stopped.\n");

}

}

// Action to pause the chat simulation

private class PauseAction implements ActionListener {

public void actionPerformed(ActionEvent e) {

if (isRunning) {

isPaused = true;

chatArea.append("Chat paused.\n");

}

}

}

// Action to resume the chat simulation

private class ResumeAction implements ActionListener {

public void actionPerformed(ActionEvent e) {

if (isRunning && isPaused) {

isPaused = false;

chatArea.append("Chat resumed.\n");

}

}

}

public static void main(String[] args) {

// Run the GUI on the Event Dispatch Thread (EDT)

SwingUtilities.invokeLater(() -> {

ChatAppletSimulation app = new ChatAppletSimulation();

app.setVisible(true);

});

}

}