**JAVA NUMBER GUESSING GAME CODE**

import javax.swing.\*;

import java.awt.\*;

import java.util.ArrayList;

import java.util.Random;

public class NumberGuessingGame {

private JFrame frame;

private int randomNumber, guessCount, minRange, maxRange;

private Random random;

private ArrayList<String> history; // To store history of guesses

public NumberGuessingGame() {

// Initialize main components

frame = new JFrame("Number Guessing Game");

frame.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);

frame.setSize(600, 500);

frame.setLayout(new BorderLayout());

random = new Random();

history = new ArrayList<>(); // Initialize history list

// Add menu bar for history

setupMenuBar();

// Show the welcome screen

showWelcomeScreen();

frame.setVisible(true);

}

private void setupMenuBar() {

JMenuBar menuBar = new JMenuBar();

JMenu historyMenu = new JMenu("History");

JMenuItem viewHistory = new JMenuItem("View Game History");

viewHistory.addActionListener(e -> showHistory());

historyMenu.add(viewHistory);

menuBar.add(historyMenu);

frame.setJMenuBar(menuBar);

}

private void showHistory() {

StringBuilder historyMessage = new StringBuilder();

if (history.isEmpty()) {

historyMessage.append("No games played yet.");

} else {

for (String entry : history) {

historyMessage.append(entry).append("\n");

}

}

JOptionPane.*showMessageDialog*(

frame,

historyMessage.toString(),

"Game History",

JOptionPane.*INFORMATION\_MESSAGE*

);

}

private void showWelcomeScreen() {

JPanel panel = new JPanel(new BorderLayout());

panel.setBackground(new Color(173, 216, 230)); // Pale blue

JLabel titleLabel = new JLabel("Number Guessing Game", SwingConstants.*CENTER*);

titleLabel.setFont(new Font("Arial", Font.*BOLD*, 32));

titleLabel.setForeground(new Color(0, 102, 204)); // Dark blue

panel.add(titleLabel, BorderLayout.*CENTER*);

JButton startButton = new JButton("Start Game");

startButton.setFont(new Font("Arial", Font.*BOLD*, 20));

startButton.setBackground(new Color(0, 102, 204)); // Dark blue

startButton.setForeground(Color.*WHITE*);

startButton.addActionListener(e -> showInstructions());

JPanel buttonPanel = new JPanel();

buttonPanel.setOpaque(false);

buttonPanel.add(startButton);

panel.add(buttonPanel, BorderLayout.*SOUTH*);

frame.add(panel, BorderLayout.*CENTER*);

}

private void showInstructions() {

JOptionPane.*showMessageDialog*(

frame,

"Welcome to the Number Guessing Game!\n\n" +

"Instructions:\n" +

"- Set a range for the number.\n" +

"- Guess the number until correct.\n" +

"- Celebrate your success with balloons!\n",

"Instructions",

JOptionPane.*INFORMATION\_MESSAGE*

);

setRange();

}

private void setRange() {

JPanel rangePanel = new JPanel(new FlowLayout());

rangePanel.setBackground(new Color(255, 228, 228)); // Light red

rangePanel.setBorder(BorderFactory.*createLineBorder*(Color.*BLACK*, 4));

JTextField minRangeField = new JTextField(5);

minRangeField.setFont(new Font("Arial", Font.*BOLD*, 16));

JTextField maxRangeField = new JTextField(5);

maxRangeField.setFont(new Font("Arial", Font.*BOLD*, 16));

rangePanel.add(new JLabel("Min: "));

rangePanel.add(minRangeField);

rangePanel.add(new JLabel("Max: "));

rangePanel.add(maxRangeField);

int response = JOptionPane.*showConfirmDialog*(

frame,

rangePanel,

"Set Range",

JOptionPane.*OK\_CANCEL\_OPTION*,

JOptionPane.*PLAIN\_MESSAGE*

);

if (response == JOptionPane.*OK\_OPTION*) {

try {

minRange = Integer.*parseInt*(minRangeField.getText());

maxRange = Integer.*parseInt*(maxRangeField.getText());

randomNumber = random.nextInt(maxRange - minRange + 1) + minRange;

guessNumber();

} catch (NumberFormatException ex) {

JOptionPane.*showMessageDialog*(frame, "Please enter valid numbers for the range!");

setRange();

}

}

}

private void guessNumber() {

JPanel guessPanel = new JPanel(new FlowLayout());

guessPanel.setBackground(new Color(255, 228, 228)); // Light red

guessPanel.setBorder(BorderFactory.*createLineBorder*(Color.*BLACK*, 4));

JTextField guessField = new JTextField(10);

guessField.setFont(new Font("Arial", Font.*BOLD*, 16));

JLabel feedbackLabel = new JLabel("Enter your guess:");

feedbackLabel.setFont(new Font("Arial", Font.*BOLD*, 16));

guessPanel.add(feedbackLabel);

guessPanel.add(guessField);

while (true) {

int response = JOptionPane.*showConfirmDialog*(

frame,

guessPanel,

"Guess the Number",

JOptionPane.*OK\_CANCEL\_OPTION*,

JOptionPane.*PLAIN\_MESSAGE*

);

if (response == JOptionPane.*CANCEL\_OPTION*) {

showExitMessage();

}

try {

int playerGuess = Integer.*parseInt*(guessField.getText());

guessCount++;

if (playerGuess < randomNumber) {

feedbackLabel.setText("Too low! Try again.");

} else if (playerGuess > randomNumber) {

feedbackLabel.setText("Too high! Try again.");

} else {

history.add("Game: Guessed " + randomNumber + " in " + guessCount + " attempts.");

guessCount = 0;

showCelebration();

break;

}

guessField.setText("");

} catch (NumberFormatException ex) {

feedbackLabel.setText("Please enter a valid number!");

}

}

}

private void showCelebration() {

JDialog celebrationDialog = new JDialog(frame, "Congratulations!", true);

celebrationDialog.setSize(400, 300);

celebrationDialog.setLocationRelativeTo(frame);

celebrationDialog.setLayout(new BorderLayout());

// Create a panel for the message

JPanel messagePanel = new JPanel();

messagePanel.setBackground(Color.*WHITE*);

JLabel messageLabel = new JLabel("You guessed it correct!");

messageLabel.setFont(new Font("Arial", Font.*BOLD*, 24));

messageLabel.setForeground(new Color(34, 139, 34)); // Green

messagePanel.add(messageLabel);

celebrationDialog.add(messagePanel, BorderLayout.*NORTH*);

// Add balloon animation

JPanel balloonPanel = new BalloonPanel();

celebrationDialog.add(balloonPanel, BorderLayout.*CENTER*);

// Add Continue or Exit buttons

JPanel buttonPanel = new JPanel();

JButton continueButton = new JButton("Continue");

continueButton.setFont(new Font("Arial", Font.*BOLD*, 16));

continueButton.addActionListener(e -> {

celebrationDialog.dispose();

setRange();

});

JButton exitButton = new JButton("Exit");

exitButton.setFont(new Font("Arial", Font.*BOLD*, 16));

exitButton.addActionListener(e -> showExitMessage());

buttonPanel.add(continueButton);

buttonPanel.add(exitButton);

celebrationDialog.add(buttonPanel, BorderLayout.*SOUTH*);

celebrationDialog.setVisible(true);

}

private void showExitMessage() {

JOptionPane.*showMessageDialog*(frame, "Thank you for playing!", "Goodbye", JOptionPane.*INFORMATION\_MESSAGE*);

System.*exit*(0);

}

static class BalloonPanel extends JPanel {

private final ArrayList<Balloon> balloons = new ArrayList<>();

private final Timer timer;

public BalloonPanel() {

setBackground(new Color(173, 216, 230)); // Pale blue

for (int i = 0; i < 20; i++) { // Increased balloon count

balloons.add(new Balloon());

}

timer = new Timer(20, e -> { // Faster animation

for (Balloon balloon : balloons) {

balloon.move();

}

repaint();

});

timer.start();

}

@Override

protected void paintComponent(Graphics g) {

super.paintComponent(g);

for (Balloon balloon : balloons) {

g.setColor(balloon.color);

g.fillOval(balloon.x, balloon.y, balloon.size, balloon.size);

}

}

static class Balloon {

int x, y, size;

Color color;

Balloon() {

Random random = new Random();

x = random.nextInt(400);

y = 300 + random.nextInt(200);

size = 20 + random.nextInt(30);

color = new Color(random.nextInt(256), random.nextInt(256), random.nextInt(256));

}

void move() {

y -= 5;

if (y + size < 0) {

y = 300 + new Random().nextInt(200);

}

}

}

}

public static void main(String[] args) {

SwingUtilities.*invokeLater*(NumberGuessingGame::new);

}

}