/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
Name: Sarah Redmon  
Date: 2/13/19  
Instructor: Ms. Tucker  
Class: Lights  
Purpose: To simulate a Simon Lights experience  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*/

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import java.applet.AudioClip;

import java.net.URL;

import javax.swing.JButton;

public class LightsPanel extends JPanel

{

/\*------------------------------------------------------------------------------  
    Initializes variables, JLabels, array, JButtons, etc. for program  
    ------------------------------------------------------------------------------  
    \*/

private JButton redButton, greenButton, yellowButton, blueButton, newGame,

rememberGame, endGame;

private Jlabel lightsHeading, newGameNote, rememberGameNote;

private int player, index = 0;

private int simonColors[] = new int[8];

private AudioClip[] music;  
    private AudioClip current;

public LightsPanel()

{

/\*------------------------------------------------------------------------------  
        Layout is set as Grid (3 by 2)  
        ------------------------------------------------------------------------------  
        \*/

setLayout(new GridLayout(3, 2));

        /\*------------------------------------------------------------------------------  
        Sound is imported and array is made to distinguish null from file  
        ------------------------------------------------------------------------------  
        \*/  
        URL url1, url2;  
        url1 = null;

url2 = null;  
          
        try  
        {  
            url1 = new URL ("file", "localhost", "Red.wav");

url2 = new URL ("file", "localhost", "Yellow.wav ");  
        }  
        catch (Exception exception) {}  
      
        music = new AudioClip[3];  
        music[0] = null;  
        music[1] = JApplet.newAudioClip (url1);

music[2] = JApplet.newAudioClip (url2);  
          
        current = null;

        /\*------------------------------------------------------------------------------  
        Heading is made as JLabel  
        ------------------------------------------------------------------------------  
        \*/  
        lightsHeading = new JLabel ("SIMON LIGHTS");  
        lightsHeading.setFont (new Font ("Comic Sans MS", Font.BOLD, 18));

/\*------------------------------------------------------------------------------  
        New & Remember Game Labels are made to show notices  
        ------------------------------------------------------------------------------  
        \*/

newGameNote = new JLabel ("");

newGameNote.setFont (new Font ("Helvetica", Font.BOLD, 20));

rememberGameNote = new JLabel ("");

rememberGameNote.setFont (new Font ("Helvetica", Font.BOLD, 20));

/\*------------------------------------------------------------------------------  
        Colors, New Game, Remember Game, & Exit Buttons are made & action listeners added into Button Listener & Color Listener  
        ------------------------------------------------------------------------------  
        \*/

redButton = new JButton ("Red Control");

redButton.setBackground (Color.white);

redButton.setPreferredSize(new Dimension(150, 150));

greenButton = new JButton ("Green Control");

greenButton.setBackground (Color.white);

greenButton.setPreferredSize(new Dimension(150, 150));

yellowButton = new JButton ("Yellow Control");

yellowButton.setBackground (Color.white);

yellowButton.setPreferredSize(new Dimension(150, 150));

blueButton = new JButton ("Blue Control");

blueButton.setBackground (Color.white);

blueButton.setPreferredSize(new Dimension(150, 150));

newGame = new JButton ("New Game");

newGame.setBackground (Color.gray);

rememberGame = new JButton ("Remember Game");

rememberGame.setBackground (Color.gray);

endGame = new JButton ("End Game", new ImageIcon ("App-x-icon.png"));

endGame.setBackground (Color.gray);

newGame.addActionListener (new ButtonListener());

rememberGame.addActionListener (new ButtonListener());

endGame.addActionListener (new ButtonListener());

ColorListener listener = new ColorListener();

redButton.addActionListener (new ColorListener());;

greenButton.addActionListener (new ColorListener());

yellowButton.addActionListener (new ColorListener());

blueButton.addActionListener (new ColorListener());

/\*------------------------------------------------------------------------------  
        Puts color buttons in ButtonGroup to better organize them  
        ------------------------------------------------------------------------------  
        \*/

ButtonGroup colorGroup = new ButtonGroup();

colorGroup.add(redButton);

colorGroup.add(greenButton);

colorGroup.add(yellowButton);

colorGroup.add(blueButton);

/\*------------------------------------------------------------------------------  
        Adds in the components to be shown in GUI  
        ------------------------------------------------------------------------------  
        \*/

add(lightsHeading);

add(redButton);

add(greenButton);

add(newGameNote);

add(yellowButton);

add(blueButton);

add(rememberGameNote);

add(newGame);

add(rememberGame);

add(endGame);

/\*------------------------------------------------------------------------------  
        Display options  
        ------------------------------------------------------------------------------  
        \*/

setPreferredSize (new Dimension (1000, 500));

setBackground (Color.lightGray);

}

private class ButtonListener implements ActionListener

{

public void actionPerformed (ActionEvent event)

{

if (event.getSource() == newGame) {

/\*------------------------------------------------------------------------------  
            If new is clicked on  
            \* Fill in newGameNote with notice

\* Empty rememberGameNote   
            \* Set player to one  
            \* Set index for array to 0  
            ------------------------------------------------------------------------------  
            \*/

newGameNote.setText("Player One: Click on 8 color buttons, one at a time.");

rememberGameNote.setText("");

player = 1;

index = 0;

}

if (event.getSource() == rememberGame) {

/\*------------------------------------------------------------------------------  
            If remember is clicked on  
            \* Empty newGameNote  
            \* Fill in rememberGameNote with notice  
            \* Set player to two  
            \* Set index for array to 0  
            ------------------------------------------------------------------------------  
            \*/

newGameNote.setText("");

rememberGameNote.setText("Player Two: Click on the correct buttons.");

player = 2;

index = 0;

}

if (event.getSource() == endGame) {

/\*------------------------------------------------------------------------------  
            If exit is clicked-closes program  
            ------------------------------------------------------------------------------  
            \*/

System.exit(0);

}

newGameNote.setFont (new Font ("Helvetica", Font.BOLD, 20));

rememberGameNote.setFont (new Font ("Helvetica", Font.BOLD, 20));

}

}

private class ColorListener implements ActionListener {

public void actionPerformed (ActionEvent event) {

/\*------------------------------------------------------------------------------  
            If certain color is clicked on

\* Set button to respective color  
            \* Play sound (different sound files) for when red or yellow is clicked on  
            \* Set simonColors[index] to certain number (ex: 1 for red)  
            \* Up the index every time a button is clicked (limit is 8 clicks)  
            ------------------------------------------------------------------------------  
            \*/

if (event.getSource() == redButton) {

redButton.setBackground (Color.red);

music[1].play();

simonColors[index] = 1;

} else if (event.getSource() == greenButton) {

greenButton.setBackground (Color.green);

simonColors[index] = 2;

} else if (event.getSource() == yellowButton) {

yellowButton.setBackground (Color.yellow);

music[2].play();

simonColors[index] = 3;

} else if (event.getSource() == blueButton) {

blueButton.setBackground (Color.blue);

simonColors[index] = 4;

}

index++;

index = index + 1;

/\*------------------------------------------------------------------------------  
            Change color button back to white when not clicked/another one is clicked  
            ------------------------------------------------------------------------------  
            \*/

if (event.getSource() != redButton) {

redButton.setBackground (Color.white);

}

if (event.getSource() != greenButton) {

greenButton.setBackground (Color.white);

}

if (event.getSource() != yellowButton) {

yellowButton.setBackground (Color.white);

}

if (event.getSource() != blueButton) {

blueButton.setBackground (Color.white);

}

/\*------------------------------------------------------------------------------  
            If player is one & index is 8

\* Fill in newGameNote with notice

Else if player is two

\* If clicked wrong, fill in rememberGameNote with losing notice  
            \* If clicked right, fill in rememberGameNote with winning notice

\* If still clicking (correctly), fill in rememberGameNote with encouraging notice  
            ------------------------------------------------------------------------------  
            \*/

if (player == 1) {

if (index == 8) {

newGameNote.setText("Now let Player Two have their turn for guessing!");

}

} else if (player == 2) {

if (index != simonColors[index]) {

rememberGameNote.setText("Oops! Try again!");

} else if (index == simonColors[index]) {

rememberGameNote.setText("Yay! You got all of them!");

} else {

rememberGameNote.setText("Keep going!");

}

newGameNote.setFont (new Font ("Helvetica", Font.BOLD, 20));

rememberGameNote.setFont (new Font ("Helvetica", Font.BOLD, 20));

}

}

}

}

for(int z = 0, j = 0; z < ctr-1; z++, j++)

{

if ( j >= n )

{

j = 0; // reset back to the beginning

}

res = (flames[j]);

jLabel1.setText(String.valueOf(res));

}