

```

// *****
//  audioPlayer.java
//  Author: Jennifer Soh  ID: JS542
//  Compiler Used: JGrasp
//  Design and implement a Java applet that simulates an audio
//  player. The applet has three buttons labeled Play, Loop and
//  Stop that makes the song play, loop, or stop.
//  *****

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.applet.*;
import java.net.URL;

public class audioPlayer extends JApplet {
    private AudioClip audioClip;
    public audioPlayer() {
        // BUTTONS CONTAINER: buttonsPanel. Set to GridLayout
        JPanel buttonsPanel = new JPanel();
        buttonsPanel.setLayout(new GridLayout(1,4));

        // CREATES THE PLAY BUTTON.
        // 1. image url, 2. image icon, 3. button 4.add to buttonsPanel

        // Play Button
        URL playURL = getClass().getResource("img/play.png");
        ImageIcon play = new ImageIcon(playURL);
        JButton playButton = new JButton(play);
        buttonsPanel.add(playButton);

        // Loop Button
        URL loopURL = getClass().getResource("img/loop.png");
        ImageIcon loop = new ImageIcon(loopURL);
        JButton loopButton = new JButton(loop);
        buttonsPanel.add(loopButton);

        // Stop Button
        URL stopURL = getClass().getResource("img/stop.png");
        ImageIcon stop = new ImageIcon(stopURL);
        JButton stopButton = new JButton(stop);
        buttonsPanel.add(stopButton);

        // Get audio location. Store it in audioClip variable
        URL audioURL = getClass().getResource("amy.wav");
        audioClip = Applet.newAudioClip(audioURL);

        // Artist Information Panel set to GridLayout.
        JPanel artistPanel = new JPanel();
        artistPanel.setLayout(new GridLayout(3,0));
        JLabel artistName = new JLabel("Amy Winehouse");
        JLabel songTitle = new JLabel("Love is a Losing Game (Demo Version)");
        JLabel album = new JLabel("Back to Black: B-Sides");

        // Adds the artist information labels to artistPanel
        artistPanel.add(artistName);
        artistPanel.add(songTitle);
        artistPanel.add(album);

        // Creates the main container for buttonPanel and artistPanel
        // Layout is set to BorderLayout.
        JPanel mainContainer = new JPanel();
        mainContainer.setLayout(new BorderLayout());
        mainContainer.add(artistPanel, BorderLayout.NORTH);
        mainContainer.add(buttonsPanel, BorderLayout.SOUTH);

        // add the mainContainer to the Frame
        add(mainContainer);
    }
}

```

```

// Adds an action listener to the playButton
// ActionListener overrides the actionPerformed to start song.
playButton.addActionListener(
    new ActionListener(){
        @Override
        public void actionPerformed(ActionEvent e){
            start();
        }
    });

// Adds an action listener to the loop button.
// ActionListener overrides action performed to loop song.
loopButton.addActionListener(
    new ActionListener(){
        @Override
        public void actionPerformed(ActionEvent e){
            loop();
        }
    });

// Adds an action listener to the stop button.
// ActionListener overrides action performed to stop song.
stopButton.addActionListener(
    new ActionListener(){
        @Override
        public void actionPerformed(ActionEvent e){
            stop();
        }
    });

}

// Defines the start method
public void start(){
    if(audioClip != null) audioClip.play();
}

// Defines the loop method
public void loop(){
    if(audioClip != null) audioClip.loop();
}

// Defines the stop method
public void stop(){
    if(audioClip != null) audioClip.stop();
}

}

```