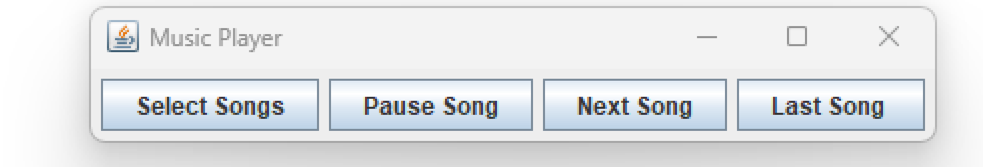
Lewis Burke

22201368

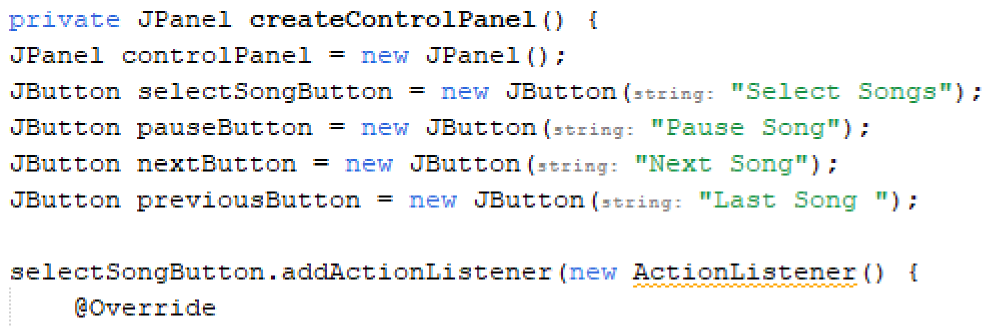
Music player -

Creating my music player proved to be a hard and tedious task as I have never used Java previously, with zero experience it felt like a big task but after hours of trial and error I gained a basic understanding. My past experience in HTML and CSS helped me slightly with the formatting of my code. Other sources like W3schools, stackoverflow and youtube really helped me understand and add new code

My player is pretty plain but I have the basic functions needed for a music player.

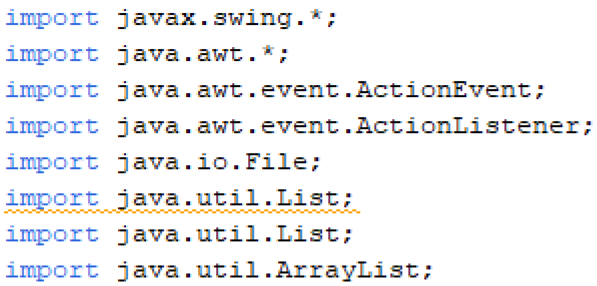


This is the code i used for my control panel



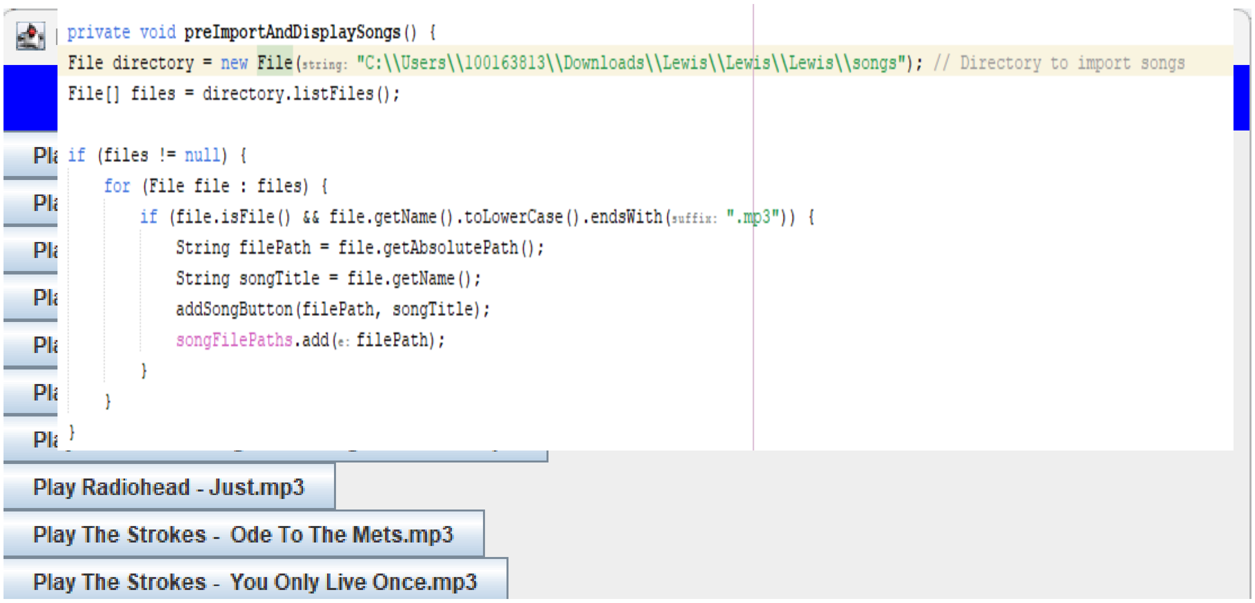
```
private JPanel createControlPanel() {
JPanel controlPanel = new JPanel();
JButton selectSongButton = new JButton(string: "Select Songs");
JButton pauseButton = new JButton(string: "Pause Song");
JButton nextButton = new JButton(string: "Next Song");
JButton previousButton = new JButton(string: "Last Song ");

selectSongButton.addActionListener(new ActionListener() {
    @Override
```



```java
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import java.util.List;
import java.util.List;
import java.util.ArrayList;
```

This is the imports i used in my GUI class, all of these imports are important to my program are important but java swing may be the most important as it is responsible for The G graphical user interface (GUI). This includes all the buttons such as ‘pause’ , the scroll wheel, etc.

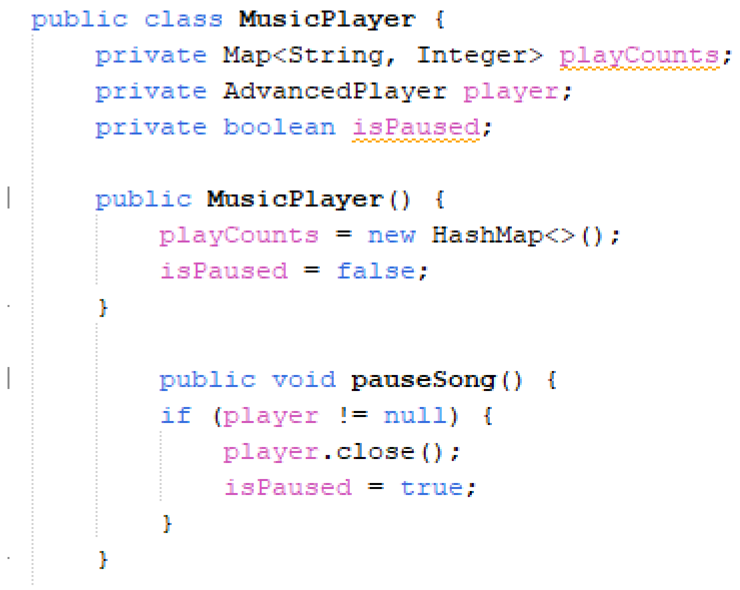


```java
private void preImportAndDisplaySongs() {
File directory = new File(string: "C:\\Users\\100163813\\Downloads\\Lewis\\Lewis\\Lewis\\songs"); // Directory to import songs
File[] files = directory.listFiles();

if (files != null) {
    for (File file : files) {
        if (file.isFile() && file.getName().toLowerCase().endsWith(suffix: ".mp3")) {
            String filePath = file.getAbsolutePath();
            String songTitle = file.getName();
            addSongButton(filePath, songTitle);
            songFilePaths.add(e: filePath);
        }
    }
}
```

Pl
Pl
Pl
Pl
Pl
Pl
Pl
Play Radiohead - Just.mp3
Play The Strokes - Ode To The Mets.mp3
Play The Strokes - You Only Live Once.mp3

Here is my player with 10 songs added to it. All of my buttons work as intended but I did run into a problem with my GUI. I struggled to add a play count, as when I added the function it would not display and I could not figure out why.



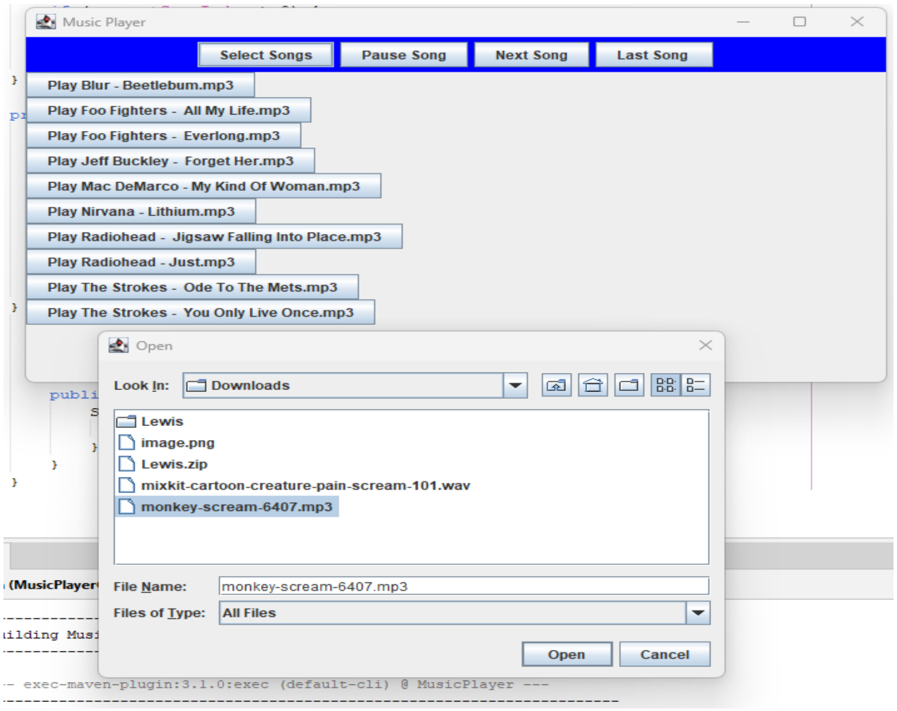
```
public class MusicPlayer {
    private Map<String, Integer> playCounts;
    private AdvancedPlayer player;
    private boolean isPaused;

    public MusicPlayer() {
        playCounts = new HashMap<>();
        isPaused = false;
    }

        public void pauseSong() {
        if (player != null) {
            player.close();
            isPaused = true;
        }
    }
```

This is the code I used to import all my songs to my player, adding songs to my file directory that mp3s are added into my player. 'Select songs' takes you into your files so you can add any mp3 on your computer.

e,g


Music Player
Select Songs
Pause Song
Next Song
Last Song
Play Blur - Beetlebum.mp3
Play Foo Fighters - All My Life.mp3
Play Foo Fighters - Everlong.mp3
Play Jeff Buckley - Forget Her.mp3
Play Mac DeMarco - My Kind Of Woman.mp3
Play Nirvana - Lithium.mp3
Play Radiohead - Jigsaw Falling Into Place.mp3
Play Radiohead - Just.mp3
Play The Strokes - Ode To The Mets.mp3
Play The Strokes - You Only Live Once.mp3
Open
Lewis
image.png
Lewis.zip
mixkit-cartoon-creature-pain-scream-101.wav
monkey-scream-6407.mp3
Open
Cancel

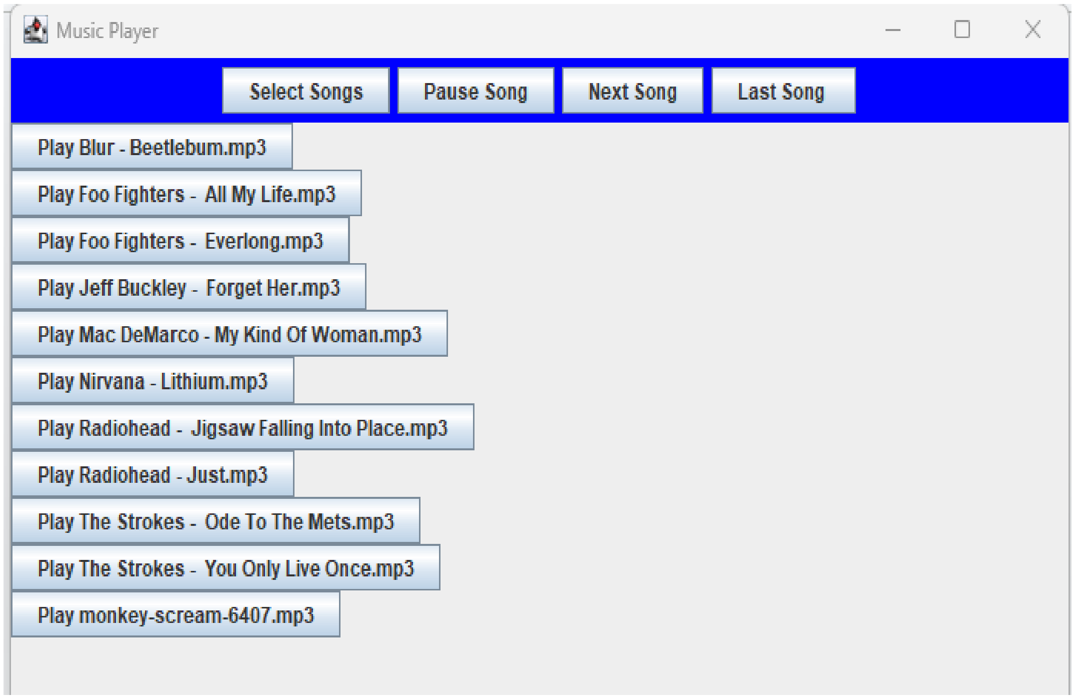
Look In: Downloads

File Name: monkey-scream-6407.mp3

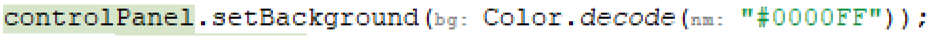
Files of Type: All Files

```
(MusicPlayer
------------
ilding Musi
------------
-- exec-maven-plugin:3.1.0:exec (default-cli) @ MusicPlayer ---
---------------------------------------------------------------
```

Here is the song added


Music Player
Select Songs
Pause Song
Next Song
Last Song
Play Blur - Beetlebum.mp3
Play Foo Fighters - All My Life.mp3
Play Foo Fighters - Everlong.mp3
Play Jeff Buckley - Forget Her.mp3
Play Mac DeMarco - My Kind Of Woman.mp3
Play Nirvana - Lithium.mp3
Play Radiohead - Jigsaw Falling Into Place.mp3
Play Radiohead - Just.mp3
Play The Strokes - Ode To The Mets.mp3
Play The Strokes - You Only Live Once.mp3
Play monkey-scream-6407.mp3

I added a background colour not to improve the design but to prove i can because i kind of like the simple colours the player loaded with



```
controlPanel.setBackground(bg: Color.decode(nm: "#0000FF"));
```

In conclusion, I feel my assignment went reasonably well. I think it is really simple but given my past experience with this type of coding, which is zero, I am not too disappointed with the result. However I think I can really improve on this, given more time. I want to improve on not only adding more functions but also improving the design aspect.

References:
[Stack Overflow - Where Developers Learn, Share, & Build Careers]
[Java Tutorial (w3schools.com)]

MusicPlayer.java class

```
import java.io.File;
import java.io.FileInputStream;
import java.util.HashMap;
import java.util.Map;
import javax.sound.sampled.AudioInputStream;
import javax.sound.sampled.AudioSystem;
import javax.sound.sampled.Clip;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import java.io.FileInputStream;
import javazoom.jl.decoder.JavaLayerException;
import javazoom.jl.decoder.Bitstream;
import javazoom.jl.decoder.Header;
import javazoom.jl.player.advanced.AdvancedPlayer;
import javazoom.jl.player.advanced.PlaybackListener;

public class MusicPlayer {
    private Map<String, Integer> playCounts;
    private AdvancedPlayer player;
    private boolean isPaused;

    public MusicPlayer() {
        playCounts = new HashMap<>();
        isPaused = false;
    }

        public void pauseSong() {
        if (player != null) {
            player.close();
            isPaused = true;
        }
    }


    public void playSong(String filePath) {
        try {
            if (!playCounts.containsKey(filePath)) {
                playCounts.put(filePath, 0);
            }

            File audioFile = new File(filePath);

            if (player != null) {
                player.close();
            }

            FileInputStream fileInputStream = new FileInputStream(audioFile);

            // Increment play count
            playCounts.put(filePath, playCounts.get(filePath) + 1);

            isPaused = false; // Reset isPaused to false before playing

            player = new AdvancedPlayer(fileInputStream);

            new Thread(() -> {
                try {
                    player.play();
                } catch (JavaLayerException e) {
                    e.printStackTrace();
                }
            }).start();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public int getPlayCount(String filePath) {
        return playCounts.getOrDefault(filePath, 0);
    }
```

Music playerGUI class

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import java.util.List;
import java.util.List;
import java.util.ArrayList;

public static void main(String[] args) {
     SwingUtilities.invokeLater(() -> {
         new MusicPlayerGUI();
     });
 }

public class MusicPlayerGUI {
    private MusicPlayer musicPlayer;
    private JPanel songListPanel;
    private int currentSongIndex = -1;
    private List<String> songFilePaths = new ArrayList<>();

    public MusicPlayerGUI() {
        musicPlayer = new MusicPlayer();

        JFrame frame = new JFrame("Music Player");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLayout(new BorderLayout());

        songListPanel = new JPanel();
        songListPanel.setLayout(new BoxLayout(songListPanel, BoxLayout.Y_AXIS));

        frame.add(createControlPanel(), BorderLayout.NORTH);
        frame.add(songListPanel, BorderLayout.CENTER);


        preImportAndDisplaySongs();

        frame.pack();
        frame.setVisible(true);
    }
private void selectSong() {
    JFileChooser fileChooser = new JFileChooser();
    fileChooser.setFileSelectionMode(JFileChooser.FILES_ONLY);
    int result = fileChooser.showOpenDialog(null);

    if (result == JFileChooser.APPROVE_OPTION) {
        File selectedFile = fileChooser.getSelectedFile();
        String filePath = selectedFile.getAbsolutePath();
        String songTitle = selectedFile.getName();
        addSongButton(filePath, songTitle);
    }
}

private void pauseSong() {
    musicPlayer.pauseSong();
}
    private JPanel createControlPanel() {
    JPanel controlPanel = new JPanel();
    JButton selectSongButton = new JButton("Select Songs");
    JButton pauseButton = new JButton("Pause Song");
    JButton nextButton = new JButton("Next Song");
    JButton previousButton = new JButton("Last Song ");

    selectSongButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            selectSong();
        }
    });

    pauseButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            pauseSong();
        }
    });

    nextButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            playNextSong();
        }
    });

    previousButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            playPreviousSong();
        }
    });

    controlPanel.add(selectSongButton);
    controlPanel.add(pauseButton);
    controlPanel.add(nextButton);
    controlPanel.add(previousButton);
    controlPanel.setBackground(Color.decode("#0000FF"));

    return controlPanel;
}

    private void preImportAndDisplaySongs() {
    File directory = new File("C:\\Users\\100163813\\Downloads\\Lewis\\Lewis\\Lewis\\songs"); //
Directory to import songs
    File[] files = directory.listFiles();

    if (files != null) {
        for (File file : files) {
            if (file.isFile() && file.getName().toLowerCase().endsWith(".mp3")) {
                String filePath = file.getAbsolutePath();
                String songTitle = file.getName();
                addSongButton(filePath, songTitle);
                songFilePaths.add(filePath);
            }
        }
    }
}

private void playNextSong() {
    if (currentSongIndex < songFilePaths.size() - 1) {
        currentSongIndex++;
        musicPlayer.playSong(songFilePaths.get(currentSongIndex));
    }
}

private void playPreviousSong() {
    if (currentSongIndex > 0) {
        currentSongIndex--;
        musicPlayer.playSong(songFilePaths.get(currentSongIndex));
    }
}

private void addSongButton(String filePath, String songTitle) {
    JButton songButton = new JButton("Play " + songTitle);

    songButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            musicPlayer.playSong(filePath);
        }
    });

    songListPanel.add(songButton);
}
```