**《宫商角徵羽》软件开发项目个人报告**

姓名：管庭威 学号：2022141461194

1. **项目基本情况**

该项目旨在开发一个功能丰富的音乐播放器，支持基本的音频播放功能，同时提供音频可视化（波形绘制）、音频剪辑和音效处理等高级功能，为用户提供一个直观、易用的音频处理工具。

**1.技术栈**

JavaFX：用于实现音频播放、波形绘制和部分用户界面。

Swing：用于构建传统的桌面应用程序界面，与JavaFX结合使用。

JLayer：用于解码MP3文件，提取音频数据。

FFmpeg：用于处理音频文件的格式转换和片段合并。

Java Sound API：用于处理音频流。

**2.项目结构**

(1)主窗口

主窗口：MusicPlayer 类负责初始化主窗口，整合各个功能模块。

布局：主窗口采用 BorderLayout 布局，分为播放控制面板、播放列表面板、歌词面板和波形面板。

(2)播放控制面板

播放控制：PlayerControlPanel 类提供播放、暂停、上一首、下一首等按钮。

进度条：提供一个可拖动的进度条，用于调整播放进度。

音量控制：提供一个滑块，用于调节全局音量。

(3)播放列表面板

播放列表：PlaylistPanel 类提供一个播放列表，支持添加文件、文件夹，以及清空列表。

右键菜单：支持右键菜单操作，如重命名、删除等。

(4)波形面板

波形绘制：AudioWaveformPanel 类负责绘制音频波形。

(5)音频编辑面板

音频剪辑：AudioEditPanel 类提供音频剪辑功能，支持分割、删除和合并音频片段。

(6)音效控制面板

均衡器：SoundControlPanel 类提供均衡器设置，支持自定义频段增益。

预设模式：支持多种预设模式。

淡入淡出：支持音频的淡入和淡出效果

1. **个人承担工作**

本人在组内承担代码编写工作，负责以下功能模块：

1. 音频播放

播放控制：支持播放、暂停、上一首、下一首等基本功能。

播放模式：支持顺序播放、循环播放（单曲循环、列表循环）、随机播放等多种模式。

进度控制：用户可以通过进度条拖动来调整播放进度。

**具体实现：**播放暂停功能的核心是通过 javafx.scene.media.MediaPlayer 类来实现的。 MediaPlayer 是 JavaFX 提供的用于播放音频和视频的类，它提供了播放、 暂停、停止等基本控制功能。

在 PlayerController 类中，play() 方法用于开始播放音频，pause() 方 法用于暂停播放；通过 mediaPlayer.statusProperty().addListener 监听 播放器的状态变化，当状态变为 MediaPlayer.Status.PLAYING 时，触发播 放事件；当状态变为 MediaPlayer.Status.PAUSED 或 MediaPlayer.Status.STOPPED 时，触发暂停事件。

2. 波形绘制

波形可视化：实时绘制音频波形，提供直观的音频可视化。

动态样式：波形绘制支持不同颜色和样式，提供美观的视觉效果。

**具体实现：**波形绘制功能通过 javafx.scene.canvas.Canvas 和 javafx.scene.canvas.GraphicsContext 类实现。Canvas 是 JavaFX 提供 的用于绘制图形的组件，GraphicsContext 提供了绘制图形的方法。

在 AudioWaveformPanel 类中，使用 javazoom.jl.decoder.Decoder 和 javax.sound.sampled.AudioInputStream 提取音频数据，在 AudioWaveformPanel 类中，使用 GraphicsContext 绘制波形。

3. 音频剪辑

分割音频：用户可以通过鼠标双击波形图来分割音频片段。

删除音频：用户可以通过鼠标右键点击波形图来删除音频片段。

合并音频：支持将多个音频片段合并为一个文件。

**具体实现：**波形剪辑功能通过 AudioTrack 类实现，它允许用户通过鼠标操作来分割和 删除音频片段**。**在 AudioTrack 类中，splitAtPixel 方法根据鼠标点击的 位置分割音频片段，deleteSegmentAtPixel 方法根据鼠标点击的位置删除 音频片段，在 AudioEditPanel 类中，mergeTracks 方法使用 ffmpeg 命令 行工具将多个音频片段合并为一个文件。

4. 数据持久化

播放状态保存：支持保存播放进度、播放模式、播放列表等信息，用户可以在下次启动时恢复之前的播放状态。

**具体实现：**播放状态的持久化主要通过 PlayerController 类中的 savePlaybackState 方法实现。这个方法将当前的播放索引、播放进度、播放模式等信息保存到 一个本地配置文件config.properties中。在 PlayerController 类的构造 函数中，通过 loadPlaybackState 方法从配置文件中加载播放状态。这个 方法在程序启动时被调用，以恢复之前的播放状态。

1. **项目问题**
2. 由于组内成员缺少沟通，导致有些新加需求没有传达到位。
3. JavaFX和Swing混合使用导致线程冲突，无法在剪辑过程中播放音频。
4. Flac格式音频过于复杂，无法处理暂停时的bug，最终砍掉。
5. 最开始使用java.awt.Graphics进行波形绘制，但精度太低于是使用javafx.scene.canvas.Canvas来完成需求。
6. **项目个人体会与评价**

这次课程体验了一个项目完整的生命周期，完成一个完整的项目是一个复杂而富有挑战性的过程，涉及到软件想法的诞生、需求分析、设计、开发、测试、部署和维护等多个阶段。

需求分析是项目成功的基础。在开始编码之前，必须与客户或用户进行充分沟通，明确项目的功能需求、性能需求、用户体验需求等。需求分析的不准确或不完整可能导致项目后期频繁变更，增加开发成本和项目风险。需求分析阶段要尽可能详细地收集需求，并通过文档、原型等方式与客户确认需求。需求文档是项目开发的“蓝图”，必须确保其准确性和完整性。

开发过程中，团队成员之间的协作至关重要。每个成员都有自己的专长和任务，通过有效的沟通和协作，可以提高开发效率，减少错误。反之，项目进程就会大打折扣。

**评价：**

优点

功能丰富、技术栈合理、用户界面友好、扩展性强、数据持久化

缺点

依赖外部工具：使用了FFmpeg进行音频处理，这增加了项目的外部依赖。如果用户没有安装FFmpeg，可能会导致功能无法正常工作。

性能问题：

波形绘制性能：对于大文件或高分辨率波形，绘制性能可能会受到影响，尤其是在实时更新时。

音频处理延迟：在进行音频剪辑和合并时，依赖FFmpeg的外部进程调用可能会引入一定的延迟。

功能限制：

音频格式支持有限：虽然支持常见的音频格式（如MP3、WAV等），但对于一些不常见的格式（如FLAC、OGG等）支持不足。

音效处理功能有限：虽然提供了基本的均衡器和淡入淡出功能，但缺乏更高级的音效处理（如混响、回声等）。