## 个人周报（软件项目管理）

**姓名**：王智豪  
**岗位/角色**：软件开发  
**项目名称**：《宫商角徵羽》软件开发项目  
**汇报周期**：2025年3月24日 - 2025年3月30日（第五周）  
**汇报日期**：2025年3月29日

### 本周工作内容总结

|  |  |  |  |
| --- | --- | --- | --- |
| **工作事项** | **完成情况** | **工作产出** | **备注** |
| 各种格式音频的播放 | 已完成 | SRS 2.0版本定稿 | 支持.wav\.mp3\.flac\.ogg格式音频的播放 |
| 异步播放控制模块开发 | 已完成 | 基于ExecutorService实现非阻塞播放 | 解决UI卡顿问题 |
| 音频格式统一化处理 | 已完成 | 实现PCM\_SIGNED 16bit格式强制转换 |  |
| 播放状态同步机制 | 已完成 | 通过volatile和Platform.runLater同步 | 确保多线程操作安全 |
| 暂停与继续 | 已完成 | 无 | 支持记录播放位置并恢复 |

### 二、工作进展详情

核心功能开发

JavaFX UI界面优化

采用BorderPane布局实现主界面分层：

顶部：文件选择栏（支持多格式音频文件加载）。

中部：播放控制按钮（动态状态切换：播放中禁用“播放”，启用“暂停”）。

底部：状态栏（实时显示播放状态及音频信息）。

按钮状态与播放逻辑绑定，提升用户体验。

异步播放控制模块

基于ExecutorService实现播放线程池管理，避免阻塞JavaFX主线程。

决因音频加载导致的界面卡顿问题。

音频格式统一化处理

强制将输入音频统一转换为PCM\_SIGNED 16bit格式，解决不同硬件设备兼容性问题。

状态同步与资源管理

使用volatile关键字修饰currentLine和state变量，确保多线程环境下状态可见性。

通过safeClose()方法统一释放音频线路资源，避免内存泄漏。

暂停与继续功能

记录暂停时的播放帧位置，恢复时通过skip()方法跳转至当前位置，确保音频帧对齐。

### 三、下周工作计划

|  |  |  |
| --- | --- | --- |
| 任务 | 预计完成时间 | 说明 |
| 开发进度条功能 | 4月2日 | 实现进度条拖动与音频跳转 |
| 集成音频波形可视化功能 | 4月4日 | 结合JFreeChart实现动态波形图 |

**个人总结：**

本周重点解决了多线程播放控制与音频格式兼容性问题，功能开发进度符合预期。

技术收获：

深入理解JavaFX多线程同步机制（Platform.runLater与volatile）。

掌握音频格式转换的核心实现（AudioSystem与AudioInputStream）。

改进方向：后续需加强异常处理鲁棒性，优化资源释放流程。