

midi 项目文档

董子扬 高俊翔 吴正博

2025.3.5

1 引言

1.1 编写目的

本文档旨在明确音乐软件项目的开发目标、功能需求及实施计划，为项目团队提供清晰的开发方向和依据。通过系统化的需求分析和功能规划，确保最终产品能够满足用户对音乐创作、编辑、协作及学习的需求，同时为开发过程中的资源协调、进度管理和质量把控提供指导。本策划书还将作为与利益相关方（如投资者、合作伙伴）沟通的核心材料，展示项目的创新性、技术可行性与市场价值。

1.2 背景

随着数字音乐创作的普及，音乐制作工具逐渐从专业领域走向大众化。然而，现有软件在用户体验、功能整合及教育友好性方面仍存在不足：

- 专业门槛高：传统五线谱软件（如 Sibelius）操作复杂，钢琴卷帘工具（如 FL Studio）缺乏直观的谱面转换功能，导致用户需多工具切换。
- 协作与分享不便：音乐创作常需团队协作，但多数工具缺乏实时协作与跨平台兼容性。
- 教育需求未被满足：音乐教育者和学生需要既能创作又能辅助学习的工具，例如通过 AI 分析错误或生成练习素材。

本项目旨在开发一款一体化音乐创作软件，整合钢琴卷帘编曲、五线谱智能转换、AI 辅助创作等核心功能，降低创作门槛，提升效率，并覆盖从专业音乐人到教育用户的广泛需求。

1.3 定义

钢琴卷帘 (Piano Roll): 通过横向时间轴与纵向音高网格可视化编辑 MIDI 音符的界面。

五线谱识别 (Sheet Music Recognition): 通过图像或手写输入自动解析五线谱内容并转换为可编辑的数字化乐谱。

MIDI (Musical Instrument Digital Interface): 数字音乐设备间通信的标准协议，用于记录和回放音符、节奏等信息。

AI 辅助作曲 (AI-Assisted Composition): 基于机器学习算法生成旋律、和声建议或自动优化用户创作。

1.4 项目内容

本软件将围绕以下核心模块展开开发：

1.4.1 钢琴卷帘编曲：

支持多轨道 MIDI 编辑，提供音符量化、力度调整、音色预览等功能。
实时生成与编辑内容同步的五线谱，支持一键导出 PDF 或 MusicXML 格式。

1.4.2 五线谱智能处理：

导入与识别： 支持扫描纸质乐谱、手写输入或图片文件，通过 OCR 技术转换为可编辑乐谱。

导出与共享： 支持导出标准五线谱文件，并生成对应的 MIDI 数据供其他软件使用。

1.4.3 扩展功能（差异化创新）：

AI 创作助手：

根据用户输入的和弦或旋律片段自动生成扩展内容。
提供乐理错误检测（如和声冲突、节奏错误）与优化建议。

教育工具包：

内置互动教程与乐理知识库。
针对学生作业提供自动批改与反馈。
通过上述功能整合，本项目致力于打造一个创作、学习、协作三位一体的音乐软件，填补市场空白，成为音乐爱好者和专业人士的“全流程创作伴侣”。

2 项目概述

2.1 项目目标

本项目的核心目标是通过技术创新与功能整合，打造一款全场景覆盖、低门槛高灵活度的音乐创作软件，解决以下行业痛点：

- 1. 降低专业工具使用门槛：通过直观的钢琴卷帘与五线谱双向编辑功能，简化音乐创作流程，让初学者和专业用户都能高效完成作品。
- 2. 打破工具壁垒：实现钢琴卷帘与五线谱的实时互转，消除多软件切换的繁琐操作。
- 3. 增强协作与教育属性：提供实时协作、AI 辅助纠错与教学工具，覆盖个人创作、团队合作及音乐教育场景。
- 4. 推动音乐创作民主化：通过跨平台兼容性与云端同步，让用户随时随地创作、分享和迭代作品

2.2 产品目标与范围

产品目标：

- 核心功能：
- 钢琴卷帘编曲与五线谱智能转换（双向互转）。
 - 五线谱扫描识别、手写输入与标准化导出（支持 MusicXML、PDF、MIDI）。
 - 多轨道混音与音色库集成（内置基础音色，支持 VST 插件扩展）。

差异化功能：

- AI 驱动创作：
- 智能和声填充、旋律生成与风格迁移（如将古典片段转为流行风格）。
- 实时乐理检测（如音程冲突、节奏错误提示）。

教育场景适配：

- 互动式乐理课程、作业自动批改与个性化练习建议。
- 教师端班级管理与学生作品集跟踪。

协作生态：

- 云端项目共享与多用户实时编辑（支持语音/文字批注）。
- 版本历史回溯与分支管理。

产品范围：

目标用户：

- 音乐创作者（独立音乐人、作曲编曲者）。
- 音乐教育者与学生（中小学、音乐培训机构）。
- 业余爱好者与内容创作者（短视频 BGM 制作等）。

2.3 应交付成果

软件产品交付：

- 可运行程序及安装包。
- 云端服务器架构（支持数据同步与协作功能）。

技术文档：

- 用户操作手册。
- 开发者 API 文档（支持第三方插件扩展）。
- 系统架构设计说明书与数据库 Schema。

测试与验收材料：

- 功能测试报告（覆盖核心功能与边缘用例）。
- 性能测试结果（响应速度、并发协作稳定性）。
- 用户反馈报告（Beta 测试阶段收集的改进建议）。

附加交付物：

- 预置音色库与 AI 模型（如和声生成模型、乐谱识别引擎）。
- 教育版专属内容包（乐理课程模板、习题库）。
- 市场推广材料（产品演示视频、功能白皮书）。

2.4 项目开发环境

开发环境：

PyCharm：主 IDE，支持 Python 多模块调试与 Git 版本控制。

Git+GitHub：代码管理与协作，采用 Git Flow 分支策略。

核心依赖库：

PyQt5：构建跨平台桌面端 GUI，实现钢琴卷帘、五线谱编辑界面。

music21：处理五线谱生成、MIDI 解析与乐理规则校验。

TensorFlow/PyTorch：驱动 AI 功能（乐谱识别、AI 作曲）。

OpenCV+librosa：支持图像处理（谱面扫描）与音频信号分析。

3 项目团队组织

3.1 人员分工

- 董子扬：组长，总体协调，技术架构，核心模块开发（如钢琴卷帘与五线谱互转逻辑）。
- 吴正博：后端与 AI 开发（算法模型、音效处理），五线谱识别模块。

- 高俊翔：前端开发（跨平台界面），用户测试，文档编写。

3.2 协作机制

- 每日站会同步开发进度，使用 Git 进行版本控制与代码审查。
- 组长董子扬负责技术方案决策，吴正博与高俊翔需在模块联调阶段紧密配合（如五线谱识别结果与前端渲染的对接）。
- 每周提交开发文档更新至共享知识库，确保技术细节透明化。

4 实施计划

4.1 风险评估及对策

技术风险

风险：五线谱识别准确率低、AI 生成音乐风格偏差、跨平台同步延迟。
对策：
采用 OCR+人工校验混合方案，优先优化常见谱面识别。
AI 模型分阶段训练，初期聚焦主流音乐风格，通过用户反馈迭代优化。
使用成熟的实时同步框架（如 Firebase），设计冲突解决机制（手动合并或时间戳覆盖）。

资源风险

风险：开发团队经验不足、项目周期紧张。
对策：
引入外部顾问指导音频引擎开发，外包非核心模块（如 MIDI 解析库）。
采用敏捷开发模式，按优先级分阶段交付（先桌面端后移动端）。

市场风险

风险：用户对 AI 功能接受度低、竞品快速模仿。
对策：
Beta 测试期提供 AI 功能的“手动修正”选项，增强用户控制感。
加速差异化功能开发（如教育工具包），建立垂直领域优势。

4.2 总体进度计划

- 需求与设计（1周）：完成用户调研、原型设计与技术方案评审。
- 技术预研（1周）：验证五线谱识别引擎、搭建跨平台框架与云端架构。
- 核心开发（8周）：
 - 前4周：桌面端钢琴卷帘与五线谱互转功能。
 - 中3周：AI助手、教育工具包开发。
 - 后1周：移动端轻量化功能与协作模块联调。
- 测试优化（3周）：分阶段测试（单元→集成→用户Beta），优化性能与体验。
- 发布运营（2周）：正式版上线，启动教育机构合作，收集反馈规划迭代。