

MIDI可视化编辑器系统设计文档

1. 系统概述

MIDI可视化编辑器是一款基于Python的桌面端音乐结构编辑软件，支持通过图形化操作编辑音符信息，并实现从图形界面到标准MusicXML/MIDI文件的转换，支持在MuseScore等工具中打开和播放。系统集成了音乐符号建模、图形界面绘制、音频播放与文件导出功能，适用于音乐教育、作曲辅助等场景。

2. 总体设计

2.1 系统架构

系统采用模块化分层架构：

- 界面层 (UI Layer)**：基于PyQt5构建图形界面，支持用户交互。
- 逻辑层 (Logic Layer)**：解析用户操作并将其转化为音符数据结构。
- 数据转换层 (Conversion Layer)**：实现从自定义结构到music21 Stream对象的转换。
- 文件交互层 (IO Layer)**：处理MusicXML/MIDI文件的保存与加载。

2.2 系统组成

- 软件部分：**
 - 操作系统：Windows 10 / macOS
 - 编程语言：Python 3.8+
 - 开发框架：PyQt5, music21, pygame
 - 支持工具：MuseScore 4
- 硬件部分：**
 - 内存：≥ 8GB
 - 存储空间：≥ 1GB 可用空间
 - 显示器：支持1920×1080及以上分辨率

2.3 数据设计

- 音符结构**：以NoteSegment类为核心，包含音高、时值、开始时间、颜色属性。
- 时间轴结构**：使用SegmentStructure合并重叠段并生成和弦/单音结构。
- 文件格式**：支持MusicXML导出、MIDI生成临时文件播放。

3. 功能设计

3.1 音符绘制设计

- 实现添加、删除、修改线段表示音符的功能
- 支持网格对齐、鼠标拖动调整音符时值

3.2 乐谱转换设计

- 使用music21构建Stream结构并导出为MusicXML
- 处理音程/和弦/休止符生成逻辑

3.3 文件保存与导出设计

- 支持手动与自动导出MusicXML文件
- 支持临时MIDI生成用于试听

3.4 音乐播放设计

- 使用pygame.mixer播放生成MIDI文件
- 添加播放控制按钮，更新播放状态

3.5 项目管理设计

- 项目名输入与保存
 - 项目内容的加载/清空/导出
-

4. 接口设计

4.1 外部接口设计

- music21接口：用于音符对象创建与MusicXML导出
- MuseScore路径：调用外部程序渲染XML（手动配置）
- pygame接口：用于播放MIDI文件

4.2 内部接口设计

- `convert_notes_to_stream(notes)`：转换自定义结构为music21 Stream
- `save_musicxml(notes)`：弹出保存窗口并导出文件
- `auto_save_musicxml(notes)`：定时保存至tmp_works文件夹

4.3 人机接口设计

- 图形界面支持鼠标拖拽、点击操作
- 日志区展示操作反馈
- 菜单栏提供快捷入口，含"新建"、"打开"、"保存"等功能

5. 性能指标设计

5.1 响应性能指标设计

- 音符操作延迟：< 50ms
- 乐谱导出耗时：< 2s（100个音符以内）
- MIDI播放响应时间：< 1s

6. 其它设计

- **可靠性**：支持崩溃自动保存，避免编辑丢失
- **可维护性**：模块职责清晰，类间耦合低
- **可扩展性**：接口预留支持更多乐谱元素，如动态、演奏技法等
- **安全性**：文件操作受限于沙盒目录（tmp_works）

附录

A. 需求/设计跟踪矩阵

正向追踪：需求规格 -> 设计

需求	对应设计模块
可视化编辑音符	3.1 音符绘制设计
导出MusicXML	3.3 文件导出设计
播放乐谱	3.4 音乐播放设计

反向追踪：设计 -> 需求规格

设计模块	支持需求
3.2 乐谱转换设计	支持音符结构化表示与XML导出
3.5 项目管理设计	支持文件管理与用户项目切换