

NotarX——音符终端 系统设计文档

1. 系统概述

NotarX（音符终端）是一款桌面端音乐创作软件，融合了钢琴卷帘等多种创作方式以及AI识别分析功能，致力于为音乐创作者、教育者与学习者提供低门槛、高交互性的创作体验。用户可以通过图形化界面编辑音符、生成乐谱，并进行MIDI试听或MusicXML导出。系统采用模块化设计，便于后续扩展与维护。

2. 总体设计

2.1 系统架构

系统划分为以下功能模块：

- 用户界面层**：实现项目管理、音符绘图、乐谱预览与交互操作。
- AI处理层**：集成AI模型，提供旋律生成、风格迁移、错误检测与个性化创作建议等功能，支持自然语言风格输入及自动谱面生成。实现项目管理、音符绘图、乐谱预览与交互操作。
- 逻辑控制层**：协调音符绘制与音符结构的映射转换。
- 数据转换层**：实现音符结构到乐谱文件（MIDI、MusicXML）之间的转换。
- 输出展示层**：负责播放音频与PDF乐谱的渲染。

2.2 系统组成

- 软件环境**：Windows/macOS系统，Python 3.8+，PyQt5、music21、pygame 等开源库。
- 硬件环境**：建议8GB以上内存、1080p分辨率屏幕。

2.3 数据设计

系统内部采用时间与音高双坐标方式记录音符，统一转换为结构化音符数据（包括单音、和弦、休止符等），支持导出为MIDI与MusicXML格式。

3. 功能设计

3.1 多格式识别与转换功能

本系统支持多种音乐格式之间的识别与互转，构建“谱-音-图”三类媒体之间的双向桥梁：

- **图片识别（五线谱图片→结构化乐谱）**：支持扫描或拍摄乐谱图片识别音符并生成可编辑乐谱。
- **MusicXML 文件解析**：加载结构化乐谱文件并呈现在图形界面中。
- **MIDI 文件逆转为五线谱**：读取MIDI生成可视化结构并允许用户再编辑。
- **MP3 主旋律提取**：分析音频主旋律并转为简化谱。

3.2 AI辅助作曲功能

- 支持用户通过旋律片段或文字描述（如“爵士”“摇滚”）自动生成对应风格旋律。
- 可实现旋律风格迁移、节奏变形等再编排。
- 内置乐理纠错与智能推荐模块，适配教育用途。

3.3 音符绘图功能

- 支持用户使用鼠标绘制音符线段，自动吸附网格。
- 提供添加、删除、移动、实时试听等交互功能。

3.4 乐谱生成功能

- 图形音符可自动转换为标准音符结构（单音/和弦/休止符）。
- 结合时值与音高规则生成标准MusicXML格式。

3.5 项目管理与播放导出功能

- 提供项目新建、保存、打开、命名等基本功能。
- 支持导出为MusicXML与MIDI文件，并自动备份。
- 支持将当前作品转换为MIDI进行播放试听。

4. 接口设计

4.1 系统接口（统一）

系统接口涵盖内部模块交互与外部工具调用两类，均以函数封装与标准格式通信方式进行整合：

- **乐谱处理**：使用 music21 进行乐谱生成、结构解析与格式转换（MIDI、MusicXML）。
- **音频播放**：调用 pygame 播放生成的MIDI文件，实现用户试听反馈。
- **文件导出与保存**：支持通过函数调用保存项目内容、导出标准文件。

- **外部工具协同**：MusicXML 输出可兼容 MuseScore 等第三方查看器或编辑器。
- **AI与图像模块接口**：预留对接AI生成模块、图像识别模块的中间层数据结构与格式协议。文件导出、PDF渲染、AI生成等模块通过函数调用解耦互通。

4.2 人机交互接口

- 图形化界面支持绘制、拖拽、点击交互。
- 提供菜单栏、状态提示栏、预览区域等典型交互元素。

5. 性能指标设计

5.1 性能指标

- 图形操作响应时间：< 50ms。
- 乐谱导出耗时：< 2s（100个音符以内）。
- 音频播放启动时间：< 1s。

6. 其它设计

- **可靠性**：自动保存与恢复机制避免数据丢失。
- **可维护性**：模块化设计便于扩展与调试。
- **可扩展性**：预留AI接口与多平台适配通道。
- **安全性**：访问路径受限于项目目录，避免误删或越权。

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 |
|----------|-----------------|
| 编辑音符 | 3.3 音符绘图功能 |
| 自动生成乐谱结构 | 3.4 乐谱生成功能 |
| 项目保存与播放 | 3.5 项目管理与播放导出功能 |
| 多格式输入输出 | 3.1 多格式识别与转换功能 |
| 智能作曲 | 3.2 AI辅助作曲功能 |