

# 作业报告

---

## 程序功能介绍

大致功能见 README.md，各页面详细功能键下面的部分类的介绍

## 项目各模块与类设计细节

### RtMidi 类

调用的外部库，是一个用于实时 MIDI 输入/输出的跨平台 C++ 库，主要用于处理音乐数据的传输和接收

### song 类

逻辑上：

将音符抽象成包含音高时值的MidiNote类，并由此派生出代表实际单音符的Note类，代表休止符的Rest类和代表和弦的MultiNote类。MidiNote类中重写了流运算符用来实现后续的保存和读取功能，并通过虚函数实现不同派生类有不同的保存格式。

将音轨抽象成包含音符对象列表和乐器音色的Track类，再将多音轨组合成的乐谱抽象成包含音轨对象列表和速度的Score类。类中同样重载了流运算符通过逐级调用下层对象实现整个乐谱的保存。

在Score类中通过将各个音轨中的音符插入以时间点为优先值的优先列表，来实现在同一个进程中播放所有音轨，这种做法相比开多个线程消除了不同音轨由于计算误差导致的不同步问题。最终实现效果是可以同时播放多种乐器演奏不同旋律。


底层实现上：

每个类中还有与RtMidi类配合的接口成员变量和调用RtMidi方法的成员函数，由此可与电脑自带的Midi端口进行通讯来播放音乐。也可以自己手动设置Midi虚拟端口，这会比电脑原生的音色更好听。

### mainwindow 类

主页面，包含读取和谱曲两个功能模块，其中读取功能可以回放之前所存储的音乐，并且显示音符音高。谱曲功能是主体功能，用于实时演奏和编辑。

### pianokeys 类

包含琴键  UI，通过键映射将键盘与琴键连接，按下或松开分别调用 NoteOn 和 NoteOff 函数控制单个 Note 的开闭以实现实时演奏。我们实现了5个八度，基本覆盖了所有歌曲常用的音域

### loadwindow 类

将储存的乐谱导入，加载到多个音轨中，实现多音轨乐谱可视化，开启新线程，通过创建 song 中 Score 的对象实现多音轨演奏

## scoreeditor 类

钢琴演奏页面，载入 pianokeys 对象，添置录制、保存等功能，通过弹奏时实时创建 Note 对象加入到对应 Track 中，从而达到标准化录制保存的目的。通过取消按键的长按转连按机制以实现延时弹奏。节拍器则通过开启新线程定时调用打击乐 Note 演奏实现

- 左端有节拍器的按钮

## editwindow 类

钢琴演奏页面衍生出的**编辑模式**，同样创建 pianokeys 对象，通过画布呈现出演奏的音调动态效果，以达到按键时间和音调高低的可视化。具体实现是通过开启多线程，引入 QTimer 计时器，在短时间内不断更新画布内容来达到动态画面的效果

- 初始线为一根红线，五秒时间从最左端移动到指定位置变为黑色，旨在为演奏提供准备时间
- 按键按下时键块颜色较为鲜亮，且可以同时按下多个键
- 左端可以调节节奏、音量等参数，并且可以自由操作添加、删除、切换音轨
- 同样包含节拍器和标准化录制的功能

## 小组成员分工情况

本项目被分为界面、功能和库三个部分，分别由组员罗、黎、朱负责，下面是具体的内容(实际工作中各有交叉)：

### 界面部分

包含几个界面的框架搭建、接口设计和琴键等 UI 效果

### 功能部分

填充几个页面各个按键的功能，并实现整个软件的多线程调配，开发了编辑界面的动态画布，使用库函数完成各个端口的连接

### 库函数部分

负责与电脑 Midi 端口的通信，抽象出了乐谱的逻辑，包含乐谱的播放，编辑(插入，删除音符，音轨，修改特定音符的成员变量)，保存与读取

## 项目总结和反思

- 本项目通过小组内配合基本实现了锚定的功能，各自发挥所长
- 本项目美术设计方面有欠缺，导致软件界面不甚美观。在界面设计上，我们主要使用了QT里比较常规的界面布局而没有进一步的美术设计