

# 北京大学 2023-2024 学年第二学期程序设计实习课程

## Qt 作业报告

组号：71      组名：Interface      小组成员：李嘉鹤、吴雨农

### 一、程序功能介绍

本项目名为“自由钢琴”，是一个模拟练琴项目。项目分为自由模式和练习模式。

在两种模式下，我们都提供了键盘或鼠标映射的长达 5 个八度的琴键，提供了两种音色，节拍器，以及录制并保存到本地的功能。在练习模式中，我们加入了富有特色并契合“练习”主题的功能。我们提供了四张初始乐谱以及两张自定义乐谱（用户可以自行录入乐谱，录入方式为同样为琴键或鼠标映射，即模拟钢琴录入）。选择乐谱后，可以自定义是否选择三种功能：显示乐谱、显示提示、显示评分。之后就可以进行练习了。当选择显示提示，练习过程中，琴键上将会有亮黄色高光提示下一个应该按什么键。值得一提的是，在这个模式中，我们对于输入进行了丰富的反馈：正确或者错误按下会触发不同的动画提示，并有醒目的变色提示。此外，左上角的统计区域将不断统计用户本次练习的得分，并在练习结束时，弹出简洁美观结算界面，根据统计数据提供趣味评级。

除此之外，我们还添加了个性化的自定义键位设置功能，使得用户可以根据自己的需要修改默认键位。键位设置将保存在本地。

## 二、项目各有效模块与类设计细节

### 1、MainWindow：程序的主窗口。

主要使用了 `QStackedWidget` 类作为页面管理器，将主页面 `HomePage`，自由模式页面 `Piano`，练习模式页面 `Practice`，设置键位页面 `SettingsPage` 这四个类的实例加入 `QStackedWidget` 中进行管理。

### 2、HomePage：主页面。

使用 `QPixmap` 和 `QPalette` 进行背景设置。使用几个 `QPushButton` 并添加槽函数来实现页面的切换。这些槽函数与 `MainWindow` 中的 `QStackedWidget` 相联系，最终由 `QStackedWidget` 实现界面切换。

### 3、Piano：自由模式。

钢琴的主体部分使用 `QPushButton` 构建完成。使用 `QLabel` 显示必要的键位、音高等提示。

使用 `QMediaPlayer` 和 `QAudioOutput` 以及声音素材完成各个声音播放器 `players` 的构建（这里我们将 `player` 设置为每次新建一个，并且设置其生命周期，可以有效防止爆破音的出现）。

使用槽函数连接 `QPushButton` 的信号与 `player`，完成鼠标映射。

对于键盘映射，我们重载了 `keyPressEvent` 和 `keyReleaseEvent`，对于特定的键盘事件进行处理，对于每一个 `key` 映射对应的 `button`，发送 `pressed` 和 `released` 信号即可。

录制功能的实现结合了多个类。`QMediaRecorder` 管理媒体录制功能，`QAudioInput` 从音频输入设备捕获音频数据，`QAudioDevice` 表示

具体的音频输入设备，QMediaFormat 定义文件格式和编解码器，QMediaCaptureSession 将音频输入连接到录音器，QUrl 设置文件路径。我们采用录制设备的“立体声混音”的输入，来检测扬声器的声音，并录制到用户指定的路径。

节拍器功能主要包括节拍器显示、节拍速度控制、节拍声音以及音量控制。节拍器显示部分包括一个显示“节拍器”字样的 RoundLabel 标签和一个显示当前节拍速度（BPM）的标签。节拍速度控制部分由一个值范围在 40 到 200 BPM 的滑块 freqSlider 和一个用于启动和停止节拍器的按钮组成，显示“开始”或“停止”。

节拍声音和定时器部分包括一个用于播放节拍音效的 QSoundEffect 对象和一个根据当前 BPM 定时触发节拍音效的 QTimer 对象。音量控制部分由一个值范围在 0 到 100% 的滑块 QSlider 和显示当前音量百分比的标签组成。在实现中，滑块的值变化会更新节拍速度和标签显示，节拍器按钮用于启动和停止节拍器，定时器按用户设定的频率播放节拍音效，音量滑块调整节拍器的音量。

音色选择功能允许用户在不同的乐器音色之间切换，以适应不同的演奏需求。音色选择按钮部分由两个 QRadioButton 对象 tone1 和 tone2 组成，分别用于选择钢琴和电子音色。当用户点击其中一个音色按钮时，会触发 onToneSelected 槽函数，该函数会根据所选音色更新每个琴键对应的音色文件路径，确保按下琴键时播放对应的音色。

#### **4、Practice：练习模式。**

与自由模式相同的功能不再赘述，这里阐述练习乐谱的内部逻辑、提示功能、结算功能、反馈。

乐谱我们使用 `QString` 将音高序列存储在 `Practice` 类中，即完成乐谱的序列化存储。我们的提示功能基于乐谱，监测乐谱 `vector` 的下标 `currentNoteIndex`，基于此进行对应按键的高亮，判断当前按下的 `button` 是否是正确的 `button`，如果是，那么更新下标，实现连贯的高亮提示。

当用户按下一个键时，程序会判断这个键是否正确。如果按键正确，系统会根据连续正确按键的数量给予不同的反馈：

1-2 次连续正确按键会显示 “Good!” 并增加 1 分。

3-5 次连续正确按键会显示 “Great!” 并增加 2 分。

6-9 次连续正确按键会显示 “Excellent!” 并增加 3 分。

10 次及以上连续正确按键会显示 “Perfect!” 并增加 4 分。

相应的反馈文字会以绿色显示，并伴随一个从按键位置向上移动并逐渐消失的动画。如果按键错误，则会显示 “Try again...”，以红色显示，并减少 1 分，当前连击次数会被重置，连击记录也会更新。错误的反馈也会伴随从按键位置向上移动并逐渐消失的动画。

在练习结束后，系统会显示一个结算界面，包括以下几个部分：

评级文本：根据用户的准确率显示不同的评级符号和颜色：

Φ（金色）：100%

V（橙色）：90%-99%

S（深橙色）：85%-89%

A（橙红色）：80%–84%

B（红色）：70%–79%

C（深红色）：60%–69%

F（深褐色）：小于 60%

统计信息：最终得分（Score）、最大连击（MAX COMBO）、准确率（Accuracy）

分类统计：“Perfect” 次数、“Excellent” 次数、“Great” 次数、“Good” 次数、错误次数

这些信息以不同的标签形式展示，字体样式和颜色根据分类的不同进行设置，使得整个界面简洁明了，便于用户查看自己的练习结果。

### **5、SettingsPage：设置键位页面。**

我们为每个键设置了对应的组件。使用 QLineEdit 读取输入（用户输入键位）。当键位设置发生变化时，触发 onKeyMappingChanged 调用 updateKeyMapping 函数，更新对应的键位映射（直接修改对应 Qt::Key 对应的 note），然后触发 saveSettings 调用 saveKeyBindings 函数，将键位设置写入本地文件。

## **三、小组成员分工情况**

李嘉鹤：节拍器，选择音色，选曲与选项，自定义练习，Score，Combo 等统计数据，正反馈动画，结算界面（共同完成）

吴雨农：钢琴基本功能与架构，自定义键位到本地，高亮提示下一个音，录制功能，外观优化，结算界面（共同完成）

## 四、项目总结与反思

1、应对突发状况的能力。在面对项目中的挑战时，团队展示出了出色的适应性。我们通过灵活调整计划和加强成员间的沟通，有效地应对了人员变动带来的影响，确保项目顺利进行。

2、查阅资料与学习能力。项目需要我们掌握多种技术，包括音频数据处理和图形界面设计。通过自主学习，我们快速掌握了必要的技术知识，保证了项目的技术需求得到满足。这一过程中，团队成员的自主学习能力和技术理解力得到了显著提升。

3、团结协作能力。此项目加深了团队成员间的合作。我们利用项目管理工具和平台，保持项目信息的透明和实时更新，确保我们都能及时获得帮助并贡献自己的力量。定期的沟通和同步让团队保持了高效的协作状态。

4、时间管理能力。在项目开发过程中，合理的时间管理对于满足紧迫的截止日期至关重要。我们通过制定明确的目标和优先级，有效地分配和使用时间资源，灵活调整进度和任务，确保关键任务优先完成。

5. 合理规划与目标设定。在项目初期，我们可能过于乐观地评估了团队的工作能力和时间资源，设定了一些难度较高的目标。这个经验教训提醒我们，在开始任何项目时都应进行全面的需求分析和资源评估，设定切实可行的目标。这不仅有助于保持项目的可控性，还能减少因目标调整带来的团队压力和资源浪费。