项目名	CoolBand
小组名	贾老师班一队
成员	曹宇森 (1900013228@stu.pku.edu.cn) 霍锦谦 (2200012652@stu.pku.edu.cn) 王子轩 (2300094819@stu.pku.edu.cn)
实现功能：	界面与选项：界面包括三部分：选项栏，乐谱栏与乐器界面：选项栏包括文件读写（未完全实现），新建、删除工程；编辑界面，主要用于乐谱编辑的设置修改；设置界面，包含乐器的按键绑定设置，乐器音量设置，节拍器开关和速度设置；乐谱界面用于可视化编辑乐谱，演奏、暂停演奏（未实现）乐谱，乐谱通过在编辑页面勾选音符长度，并在乐器界面演奏来实现编辑；乐器界面主要负责乐器的演奏（包括鼠标点击和键盘两种交互）
	演奏：在乐器界面，通过点击乐器的相应方位（钢琴的琴键，架子鼓的军鼓、镲，底鼓等）播放对应乐器，也可通过按下键盘对应按键演奏：乐器包括：架子鼓，钢琴，待补充（未实现）
	编曲：模拟乐谱，功能包括：添加、删除、修改音符，修改时值（例如，一分钟 120 拍、90 拍），选择节奏型（4/4 拍，3/4 拍），多个乐器同时编辑（演奏部分未实现）；

项目模块与类设计细节：	乐谱模块 scoreSection 类：不同乐器的乐谱模块通过 QHBoxLayout 在可视化界面上进行管理，乐谱模块主体为继承 QWidget 的子类。scoreSection 类中包含的控件由一个 QVBoxLayout 进行管理，包含的控件包括： ● QLabel，用于显示乐器名称 ● QSpinBox，用于调整乐谱的 bpm ● QCheckBox，用于改变乐谱是否处于编辑状态 ● QPushButton，三个，分别用于：播放，暂停，从头开始播放乐谱 ● QGraphicsView，乐谱可视化界面，用于显示添加的音符 scoreSection 中声明了多个类： 1. note 类,为音符的数据表示，实际上即是一个整数对（时值，音符）。 2. bar 类,为乐谱中一个小节的数据表示，包含参数有：int remainingLength，用于计算小节剩余的可添加的时值，vector noteInBar 用于储存 note；以及 noteInBar 的 iterator。 以上两个类主要用于在演奏乐谱内容时提供乐谱上音符的时值和键值。

	3. graphicNote 类，主要用于：在给定时值和键值时，生成一个该音符的*简谱*表示，继承于 QGraphicsItem，主体为 QGraphicsItemGroup，生成时整合下划线（时值的表示），点（音高表示），#号（是否升半音）等的对应 QGraphicsItem，通过 createGraphicsItemGroup 函数生成一个整体的 QGraphicsItem，加入到 QGraphicsView 中。 4. MyGraphicsSimpleTextItem 类，继承于 QGraphicsSimpleTextItem，唯一区别在于：重载了构造函数，使构造更简单，且改变了默认字体和颜色； scoreSection 类为了演奏乐谱，还包含一个 vector<bar*> bars，用于记录小节（和小节中的音符信息）；为了更方便的管理、删除 graphicNotes，还有一个 vector<graphicNote*> graphicNotes。 scoreSection 和其他模块的联动： ● 通过激活乐器按钮，（并设置音符时长）可以在乐谱上添加音符； ● 通过按下 backspace 按键，可以删除乐谱上的音符。

	MySoundEffect 类：继承于 QSoundEffect 类，主要目的是：QSoundEffect 类在执行 play()函数时，若已经处于播放状态，则当前的播放会被强制终止。对于乐器而言，这样是反直觉的。所以该类重载了 play()函数，在调用函数时，若当前正处于播放状态，则生成一个 source，volume 相同的 QSoundEffect 指针，并调用该指针指向的对象的 play()函数。
	keySettingDialog 类:主要用于 keySettingAction 中。keySettingAction 主要用于改变乐器按键绑定的键盘按键。如果用 QLineEdit 等的方式输入，可能需要较复杂的方式处理输入的内容。于是便用这种方式代替:在按下编辑按钮时生成 QDialog 对象，捕捉键盘事件，并将按键值赋值给乐器按钮对应按键值。
	乐器模块： 1. 钢琴部分:钢琴主体为用 QButtonsGroup 批量管理的 QPushButton，通过 setGeometry(),raise(),setStyleSheet()形式模拟钢琴的视觉效果，钢琴的 QSoundEffect 由 vector<QSoundEffect*> pianoEffects 管理 2. 架子鼓部分：架子鼓主体为 QPixmap 设置的图

	片，按钮和钢琴部分一样由 QButtonsGroup 管理，QSoundEffect 由 drumEffects 管理。 3. 节拍器：主体为 QTimer，通过计数的方式实现强音和弱音 4. 键盘控制：通过重载 keyPressEvent()实现。
	设置部分（音量、按键、节拍器等）：主体是自己编写的 QDialog，手动添加 layout、控件、绑定控件信号的槽函数。
分工情况	王子轩：负责了 scoreSection 主体的编写，乐器主体可视化部分的编写吗。界面 ui 设计。 曹宇森：提出构思；乐器的槽函数（后期由王子轩添加与 scoreSection 的联动），设置界面 QDialog（和 keySettingDialog 类）的设计和编写（1/3）。 霍锦谦：资源管理，包括音效、图像资源，节拍器部分，设置部分 QDialog 的编写（2/3）。
感悟	1. 当需要大量对某个类进行修改（尤其是不可/很难进行脚本化修改，即通过循环修改）时，不妨可以试一下继承这个类，并自己编写、重载成员函数，以简化工作 2. 在编写代码量巨大的类时，需要对变量进行管理 3. 在工作初期，尽可能的规划好变量类型/管理方式，提出可能出现的问题。避免之后工作因为管理

	方式的效率低下，需要大量修改前期代码。 4. 在开始工作签，对可以使用的资源进行了解（例如控件的功能、效用等） 5. 尽量将工作模块化。