

基于 Qt 的个性化音乐播放器的设计与实现

程御风 白俊鸽

四川大学锦城学院 计算机与软件学院 四川 成都 610000

【摘要】现如今音乐播放器种类繁多，而界面设计技术的发展也让音乐播放器的界面变得更加美观和方便。然而市面上开发的音乐播放器一方面是满足了用户的使用体验和视觉美观，但是另一方面又为了符合商业化的利益需求，增添了很多商业广告和歌曲推广之类的内容。本文以音乐播放器为例来介绍怎样利用 Qt 图形开发框架来设计和实现一个能够满足用户音乐欣赏的简洁的个性化音乐播放器，通过基于 C++ 开发技术，利用 Qt 应用程序开发框架可以满足用户需求的应用程序。使用 Qt 框架不仅可以更简便的进行界面设计，而且可以使程序占用内存更小，使用更加方便。

【关键词】音乐播放器；界面应用程序开发框架；Qt 界面

1 引言

随着音乐平台不断的合并兼容，音乐播放平台开始变得越来越商业化，而人们在日常生活中经常需要用音乐来宽慰心灵，排解寂寞，娱乐生活，所以一款心仪的用来欣赏音乐的音乐播放器就显得尤为重要，这也要归功于音乐播放器将音乐文件进行整合，然后以列表的形式展现出来，供人们欣赏使用。而如果能够拥有一款个性化的私人定制的音乐播放器，既可以使人们享受到音乐，又可以避免受到令人心烦的广告和推荐的骚扰，无疑可以令很大部分人感到方便和舒服。对于此，一款干净简洁并且可以个性化的音乐播放器便具有以上优点。

2 技术选型

对于选择哪种界面开发框架时，我们首先需要考虑的是开发程序的便捷性。先看基于 MFC 的界面设计开发框架，由于 MFC 是微软公司提供的的一个类库，它是对大部分 Win32 API 函数进行了一个以 C++ 形式的封装，并且包含一个应用程序框架，其中还包含了非常多的图形环境应用程序的框架，还有 Windows 的内建控件和组件组件^[1]，其提供的这些功能使用起来虽然确实比较的方便和快捷，但是由于其语言使用固定的结构，接口繁多，查阅文档也不是很方便，故会导致学习周期长，开发量大，界面的美观设计起来也不是特别方便。

而基于 C++ 的界面开发框架 Qt，它的框架内的语言是直接采用 C++ 编写的，对于大部分编程人员都很容易掌握其使用，学习周期短，只需要弄懂其中的核心特征——信号与槽机制^[2]即可，其具有大量的开发文档，可以在开发时就直接进行查询，即查即用，极其方

便，并且它的界面部件很多种类，可直接在 UI 中设计好界面，再在框架中实现文本等属性设计，因此界面美观设计十分容易实现。并且 Qt 框架有着一个最大的优点，就是它具有十分优良的跨平台性，Qt 支持的操作系统有 Linux、Microsoft Windows 95/98、Microsoft Windows NT 等主流操作系统，能够实现一次编程，处处可用，所以 Qt 的移植性十分优良。

3 主要功能描述

由于此音乐播放器是本地的个性化应用，面向的是开发者或者使用者自己，因此应用程序不需要设计网络和登录等功能。根据需求实际设计出的功能可以概括为三个部分，包括歌曲列表初始化导入和新增功能、音乐播放暂停等基础操作功能和歌词文件自动检索并显示功能。

用户能够通过自定义的初始音乐文件路径，将喜欢的音乐放置在此路径之下，使用程序时便可以看到所有喜欢的歌曲，并且可以通过扫描本地音乐文件功能来添加其他音乐到列表中。歌曲的基础功能包含有歌曲的暂停、播放、切换到上曲或下曲、音量增加和减少，以及音乐的播放模式的转变。歌词自动检索显示功能可以在指定的目录下自动检索当前播放的音乐有无歌词文件，从而进行实时显示歌词或者提示无歌词文件。

4 界面设计

4.1 音乐播放器主窗口

音乐播放器主窗口分为三个内容区域，包括主窗口信息区域、歌曲列表显示区以及功能按钮和基本信息区。

在主界面标题栏显示自定义的音乐图标和程序名, 主体区域用来显示通过绑定函数显示当前音乐文件的歌词。歌曲列表展现可以播放的歌曲名称, 在歌词显示区域会展现当前的播放歌曲的歌句。通过功能按键和基本控件就可以实现音乐文件的操作和相关信息显示, 例如当前歌曲文件的总时长, 正在播放的歌曲的名称, 歌曲播放的位置通过进度条显示等, 如图 1 所示。



图 1 音乐播放器效果

4.2 歌曲列表显示区

歌曲列表显示区主要是显示当前路径下的所有音乐文件, 然后还可以通过扫描本地音乐文件功能按钮来实现歌曲列表的增加, 并且可以对想播放的音乐进行鼠标双击操作, 然后程序就会通过绑定的信号与槽函数来实现选中文件的播放。

4.3 功能按钮和基本信息区

功能按钮和基本信息区分为两类, 功能按钮类显示了可以进行的操作的功能按键, 包括了播放暂停功能、上下曲功能、调节音量大小功能, 此外为了方便用户使用体验, 还设计了进度条拖动改变音乐播放位置功能和播放模式功能, 此音乐播放器默认的播放模式为当前列表中循环播放, 可通过模式切换功能按钮切换为单曲循环或者随机播放模式。第二类是歌曲基本信息显示类, 包括进度滑动条实时显示的歌曲播放进度、标签显示的当前播放歌曲时间位置和歌曲总时间信息、显示当前播放歌曲的名字, 还有音量滑动条显示当前音量值。

5 具体功能模块实现

5.1 主窗口的歌词显示模块

主窗口是一个 Qt 的基础窗口部件 QWidget 实现的, 在标题栏实现音乐播放器图标和名称, 接着在主窗口上新增一个 QWidget^[3], 再增添五个 label 来显示五行歌词。具体实现是分为三步, 第一步需要使用 QFile 来读取用户给定歌词文件所在的文件路径, 其中使用一个 QFile 类的用于读取文件和写入文件的 API, 当读取到

歌词文件后, 通过自定义的正则表达式 `const QRegExp rx("\\(\\d+):(\\d+(\\.\\d+)?)\\)\\")` 来查找时间标签的正则表达式, 然后通过解析将时间标签和歌词文本放入以一个自定义的包含时间和文本的 class 类的 vector 中。自定义的 Class 类: `Class LyricLine{public: qint64 time; QString text; LyricLine(qint64 time, QString text): time(time), text(text) {}}`; 储存所有歌词的 vector: `vector<LyricLine> lines`。第二步就是通过 Qt 的 QMediaPlayer 的播放位置属性 &QMediaPlayer::positionChanged 与歌词时间标签对比来定位当前播放位置, 形成歌词行号索引, 在设置一个获取函数通过获取索引值来从 vector 中获取歌词文本。然后就是第三步, 将获取索引值通过对比播放位置将歌词文本值显示在设置的 Widget 的五个 label 上, 设置连接信号和槽函数, `connect(Player, &QMediaPlayer::positionChanged, this, &MyWidget::updateSongLrc)`; 通过播放位置的不断改变就实现了随时间动态变化歌词。

5.2 功能按钮和基本信息模块

首先是实现歌曲的播放和暂停功能按钮, 这里首先定义了一个布尔类型变量, 初始化为 false, 然后在定义的播放槽函数中对这个 bool 值进行判断, 为 false 时就通过 `Player->play()` 来实现播放, 否则就利用 `Player->pause()` 来实现暂停, 关键要定义连接机制来实现点击播放按钮后就会发出已经按下此按钮的信号来调用定义的播放槽函数, 连接代码为: `connect(ui->PlayButton, &QPushButton::clicked, this, &MyWidget::on_PlayButton_clicked)`; 然后就是上下曲功能, 这个要通过初始化的 QMediaPlaylist 索引值变化来实现, 直接上一曲就通过调用 `Playlist->previous()` 可以实现, 下一曲就是 `Playlist->next()` 实现, 连接机制也同上方法。切换播放模式就可以通过 Qt 文档中的 QMediaPlaylist::PlaybackMode 的值来改变, 具体的有 `CurrentItemOnce`(当前歌曲播放一次)、`CurrentItemInLoop`(当前歌曲循环)、`Sequential`(歌曲列表顺序播放)、`Loop`(歌曲列表循环播放) 以及 `Random`(随机播放) 这四种。对于音量值设置这个也可以直接通过内置函数 `Player->setVolum(value)` 进行音量改变, 但如果要通过拖动进度条来改变音量, 就要先设置一个 Slide 进度条, 通过获取进度条的值, 将其赋值为 `setVolum(value)` 中 value 的值, 就可以达到改变音量的效果。最后就是音乐播放进度条显示实时播放进度与通过拖动进度条来改变音乐播放位置, 这个就要同歌词一样先实时获取播放位置, 再通过 `setRange(0, Player->duration())` 设置进度条长度值范围, 并将获取到的播放位置 position 传给进度条的 value; 就可实现进度条随着播放改变随着改变, 而通过拖动进度条改变歌曲位置就是相反着的, 将拖动到的位置通过 `connect(ui->SliderDu`

ration,&QSlider::sliderMoved,Player,&QMediaPlayer::setPosition) 赋值给播放的歌曲位置就能够实现通过进度条位置改变来更改歌曲播放位置的功能。因为程序中时间是毫秒制的,所以要显示当前播放的时间和时间标签就要先自定义一个时间转换的函数,将播放时间转换为我们能读懂的分秒形式,就可实时显示时间了^[4]。而显示当前播放歌曲名就是设置一个 `index=PlayList->currentIndex()` 来获取当前播放位置,再通过 `QString songlabel=fileList.at(index);` 来提取在当前位置的文件名,最后将这个文件名设置为 `label` 上的文本就完成可以完成显示了。

5.3 歌曲列表模块

在歌曲列表这里,首先使用到的是一个 `QWidget` 控件,这个 `tableWidget` 是可以显示出数据表格的控件,其用来显示歌曲列表就十分的合适,其中的单元格数据可以通过 `QWidgetItem` 的对象来实现填入,并且在此中可以通过更改单元格属性来实现有无单元格框,是否有表格头,单元格内文本颜色等,这里为了美观,将单元格背景设置为透明,使歌曲列表浮于主界面。然后在全局中要先定义一个 `QStringList` 类的文件列表 `fileList`,用来储存读出的文件名,再通过创建 `QWidgetItem` 的对象来将数据全部填入到表格中,然后将每个歌曲文件通过 `PlayList->addMedia(QUrl::fromLocalFile(Path)` 这个 `QMediaPlayer` 歌内置函数添加到歌曲播放列表,这样歌曲列表界面就可以显示在歌曲列表中的歌曲并且实现播放了,为了方便用户的使用体验,还增加了通过鼠标双击歌曲名来选中列表中的歌曲进行播放,实现代码为: `int selectdeindex=item->row();PlayList->setCurrentIndex(selectdeindex);` 这个实现原理就是设计一个槽函数,函数中通过将获得选中的行号设置为当前播放位置,通过发出 `doubleclicked` 信号到此槽函数中,

就可实现此功能。在列表下方还增加了一个扫描本地歌曲按钮,通过定制 Qt 中的 `QFileDialog` 标准文件对话框类,实现选中本地任意指定路径音乐文件的功能,再将文件添加到 `fileList` 列表中,通过前面设置的显示到 `QWidget` 功能,就可以实现列表的增添文件功能。

6 结束语

本文介绍 Qt 界面设计开发框架的使用,基于此设计的个性化音乐播放器已经实现,并且可以在 Windows 操作系统上正常使用,实现了基本的播放、暂停、歌曲切换、双击列表播放、歌词自动检索显示、播放模式切换以及时间和音量进度条功能,还有进行了界面背景个性化定制,图标的自制,文本颜色等的自定义等个性化设置。因为此音乐播放器是基于 Qt 框架开发的,所以只要将此代码移植到其他平台编译,照样能够正常运行。此款音乐播放器不仅实现了个性化美观、简洁、方便使用的优点,还具有特别好的跨平台特性,一次开发就可以在各平台编译,所以在 Qt 开发软件拥有开发的成本小、开发拓展性强、开发的时间周期短暂等特点,且代码简单易懂。基于丰富的接口、文档以及优越的跨平台性,相信在将来 Qt 很可能会成为主流的界面设计开发框架。

【参考文献】

- [1] 徐璇,姜明新,黄静,徐晶,李敏.基于 MFC 的工程软件界面设计[J].电子设计工程,2011,19(21):11-13.
- [2] 龚凌璞.Qt 框架中信号和槽机制的研究[J].计算机光盘软件与应用,2013,16(11):281+283.
- [3] 芦东昕,周建彬,谭振华.基于 Qt/Embedded 的控件扩展研究与实现[J].计算机技术与发展,2006(10):97-100.
- [4] 焦正才,樊文侠.基于 Qt/Embedded 的 MP3 音乐播放器的设计与实现[J].电子设计工程,2012,20(07):148-150.