

基于 QT 的音乐播放器的设计与实现

杨璐宁 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】如今,市场上的音乐播放器各具特色且资源丰富。但是,也存在着一些不足,主要问题有:商业广告多,占用资源多、操作复杂等。本文通过设计开发了一款基于 QT 的音乐播放器,以个性化、简洁、能满足大多数人需求为特点,带给用户良好的使用体验。

【关键词】QT; 音乐播放器; C++

随着网络的发展,人们可以通过浏览器、音乐播放器软件等方式获取音乐资源,但通过浏览器搜索可能因为用户不是会员或权限不足无法播放音乐,通过音乐播放器则会经常收到广告、邮件,甚至突然出现弹窗给用户的使用带来影响。同时因为软件功能的复杂,在后台运行了本不需要的进程,占用很大的内存^[1]。为了解决这些问题,本文利用 QT 的界面框架,基于 Visual Studio 的开发环境,实现了播放器的暂停、播放、上下曲切换、歌词显示等功能,设计了一个操作简单、界面友好、支持多种音乐文件格式、占用内存小的音乐播放器。

1 开发技术的选择

1.1 Qt 类库

由于跨平台的软件在移植到另一个平台时可能会出现报错,而 QT 可以通过 qmake 工具,能很好地支持跨平台构建,并且 QT 是一个企业基于 C++ 的图形用户界面的开发研究框架,它提供了一套整合的开发工具、丰富的应用程序接口函数以及多达 250 个以上的 C++ 类。相比于 MFC 等类库,Qt 支持的风格样式更多,用户使用界面设计工具也更加方便,面向对象的特征明显,这也是选择 Qt 进行界面编程的主要原因。本次的设计主要用到的 Qt 的基类有:QObject、QTime、QMediaPlayer、QWidget、QString^[2]。

QObject 类:QObject 及其子类都可以通过使用 connect 建立一个信号-槽的连接。

QTime 类:对于用户来说提供了一定的时间功能,用来控制时间。项目中初始化了 QTime 对象,可以直接进行时间设置,从而获取音乐播放的时间。

QMediaPlayer 类:用来实现网络或本地音视频的播放,QMediaPlayer 中提供了信号、成员函数、公有槽,通过它们完成对音乐文件的操作,比如播放、暂停、进

度改变等。项目中用到该类的函数主要有:position() 获取当前音乐的播放位置;duration() 获取当前选中音乐的总时长;setVolume() 设置媒体声音。

QWidget 类:窗口类,QLabel 和 QLineEdit 等界面窗口部件都继承于它,它的功能是从窗口接收事件并在屏幕上绘制以及处理用户输入。新生成的窗口需要调用 show 方法才出现。

QString 类:表示一个字符串类,它采用了 Unicode 码去存储数据字符,每一个字符是 16 位的 QChar,所以可以用它来处理中文字符。项目中用到该类的函数主要有:compare() 比较两个文件的地址是否相同;number() 实现数字与字符串的转换。

1.2 信号-槽机制

信号-槽机制在 Qt 中是很重要的一部分,是 Qt 框架中所特有的^[3]。它应用于对象间的通信,当某一个对象的状态改变,这个对象就将信号发射出去。槽用来接收信号,但槽不知道自己与哪一个信号相连,这种方式使各组件互相独立。信号-槽是能够代替回调的技术,虽然比回调函数的方式慢,但它的处理方式更加地灵活,降低了 Qt 对象之间的密切关系程度。当信号-槽关联上后,就保证了槽得到了调用,可能在运行时将关联的对象删除掉,应用程序也会正常运行。本次的设计用到的信号和槽连接的方式有:connect() 基于目标函数进行指针的重载形式、信号和槽的自动关联。

connect() 基于目标函数进行指针的重载形式:这种方式是 Qt5 中加入的,槽函数不用必须是 slot 关键字进行声明的函数。在程序中的应用是:connect(&player.&QMediaPlayer::positionChange,this,&musicplayer::musicPosChange),将成员函数 musicPosChange 与信号连接,实现音乐播放时进度条的移动。

信号和槽的自动关联:以 on_对象名_信号的形式

式命名槽函数,直接实现信号和槽的连接,不必再写 connect 函数了。在本项目的增加删除、播放暂停等模块都采用了这种方式。

1.3 C++ 简介

C++ 比 C 语言更加高级,是一种编程语言且面向对象,因此它具有面向对象语言的四大特性。标准的 C++ 的组成部分有:核心语言,它提供了几乎所有的构件;C++ 标准库,它又包含标准函数库和面向对象类库,前者包括输入/输出、字符串处理、时间日期等,后者包含 C++ I/O 类、String 类、STL 容器类、数值类等类及其相关函数。因为面向对象的这一特点,与其他高级语言相比,C++ 的处理运行速度最快,使用也非常灵活,因此许多游戏软件和系统都由 C++ 来编写。本次的项目开发语言选择了 C++ 语言,在 Visual Studio 编译环境进行本地配置,然后在 Visual Studio 上创建 Qt 项目。

2 功能需求分析

目前存在的播放器多种多样,但是很多音乐播放器不能将音乐文件显示在界面上,而对于一个音乐播放器,最基本的功能要能播放音乐文件。在播放音乐的过程中,用户可以根据需求增加一个或多个本地的音乐文件到播放列表,能根据需求进行播放暂停、上下曲切换、增加删除文件、调节音量大小、显示歌词、显示音乐播放时长、显示歌曲背景图片等,而且音乐文件的格式有多种,所以这款音乐播放器不仅要满足用户的各种需求,还要能支持多种音乐格式。

2.1 功能实现

2.1.1 UI 界面设计

成功创建 Qt 项目后,在项目文件中创建三个文件夹,分别用来存放歌曲、歌词、背景图片以及按钮样式图片。然后点击 ui 文件,进入 Qt 设计工具中并进行运行界面的设计。本音乐播放器的 ui 设计主要分为四个部分:

添加 QPushButton 控件制作按钮,包括添加音乐按钮、删除音乐按钮、播放\暂停按钮、上一曲\下一曲按钮、静音模式按钮,其中播放\暂停、静音模式、上一曲\下一曲实现图标按钮的样式。

添加 QListWidget 控件分别制作歌曲文件列表和歌词列表。

添加 QSlider 控件制作两个水平的滑动条,分别实现音乐播放进度滑动条和声音大小控制。

添加 QLabel 控件来实现音乐总时间、当前播放进度时间和音量大小百分比的显示。

2.1.2 具体函数实现

本播放器定义的函数以及实现如下:

1) 槽函数 void on_pushButton_openorpause_clicked():

点击图标按钮后,实现音乐播放或暂停的函数,而且按钮样式会随着点击而改变为播放或暂停的状态。

2) 槽函数 void on_pushButton_add_clicked(): 首先使用 getOpenFileNames() 可以选择在音乐列表增加一个或多个歌曲文件,再用 for 循环判断列表中是否有相同地址的歌曲,如果歌曲不重复则可以添加,获取路径中的文件名添加到音乐列表。

3) 槽函数 void on_pushButton_del_clicked(): 用 QListWidgetItem 中 takeItem() 删除列表中当前选中的音乐文件,后面的文件再依次前移。

4) 槽函数 void on_pushButton_last_clicked(): 点击上一曲按钮后,切换到相应的歌曲并重新加载该歌曲的背景图片和歌词。若列表只有一首歌曲,点击按钮后重新播放该歌曲。

5) 槽函数 void on_pushButton_next_clicked(): 原理与 on_pushButton_last_clicked() 相同,点击下一曲按钮后自动播放下一首歌曲。

6) 槽函数 void on_listWidget_list_doubleClicked(const QModelIndex& index): 双击列表中某一歌曲文件后,播放该歌曲以及显示歌词,调用 WindowBackgroundImage() 显示背景图片。

7) 槽函数 void on_pushButton_voice_clicked(): 当音乐播放时,点击声音图标按钮后,图标切换为静音样式,声音为零。

8) 槽函数 void on_horizontalSlider_volume_valueChanged(int value): 通过移动滑动条改变音量大小的函数,并且使用 QLabel 中 setText() 显示当前音量的百分比。

10) 槽函数 void on_horizontalSlider_musicitime_valueChanged(int value): 移动音乐播放进度条实现音乐快进或快退的函数。

11) 槽函数 void musicPosChanged(qint64 position): 音乐播放时,进度条随着时间移动,找到对应时间的歌词,并且当前歌词在列表中放大显示,实现歌词字幕滚动条。可以通过改变参数 position,来实现歌曲快进和快退。当前音乐播放完毕,自动跳转到下一首。此函数使用 Qt 中 QIODevice 类来管理输入和输出设备,对于文件操作,选择以只读的方式打开。

12) 函数 void WindowBackgroundImage(): 从指定路径中加载相应歌曲背景图片的函数,调用 QPixmap 类中 scaled() 使图片大小拉伸自动适应窗口。

13) 函数 void keyBoardEvents(QKeyEvent* e): 选中音乐列表的某一首歌曲,按下回车键便能播放该歌曲。

2.1.3 支持多种音乐格式实现

为了使音乐播放器可支持多种文件格式的音乐,需要安装解码器。常用的解码器有 FFDSHOW,但它缺少相

应的分离器对于一些比较新或罕见的音视频文件格式。还有一些解码器价格比较高,而且集成度低、程序兼容性低、没有一个统一的方式处理,而 LAV Filters 的出现弥补了大部分的缺陷。LAV Filters 几乎可以播放任何格式音乐文件和现代媒体。它是一种基于 ffmpeg 项目中的 libavformat/libavcode 库,是一套具有开源的 DirectShow 分离器和音视频解码器,被评价为目前功能最齐全的 DirectShow 插件。它的音频解码器和视频解码器封装了 libavcodec。在本地进行下载需要安装的 LAV Filters 后,在 Qt 环境中进行音乐播放器开发,还可以解决无法播放音乐、播放卡顿等问题。

3 结束语

本文介绍了如何用 Qt+Visual Studio 设计出一款音乐

播放器,探讨了 Qt 框架的优势、如何将信号-槽机制应用在实际项目中、Qt 中常用基类的区别和作用等,分析了 LAV Filters 解码器的功能。本音乐播放器基本实现了用户所需要的功能,因为是在 Windows 系统下集成的 Qt 环境,所以只需要将代码在其他 Qt 能运行的平台上进行编译,就能在当前平台运行了,达到了简单方便的目的。

【参考文献】

- [1] 朱光洪. 基于 MFC 的视频播放器的设计与实现 [D]. 长沙: 湖南大学, 2017.
- [2] 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.
- [3] 朱浩悦, 张红祥. 基于 QT 的视频媒体播放器的设计与实现 [J]. 科技信息, 2011(30): 274-275.