

基于 QT 的 MP3 音乐播放器设计和实现

倪誉丹 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】音乐播放器一直是人们离不开的娱乐软件之一，现在市面上存在着很多不同种类的音乐播放器，这些播放器功能一步步完善的同时，也带来的许多问题：界面过于繁杂，各种推广广告太多。开发者从这两点出发，设计实现了基于 QT 的音乐播放器^[1]。开发采用了 Visual Studio 2017，基于 QT 实现了打开音乐、播放音乐、暂停和继续、播放上/下一首、设置音量、设置播放模式、打开播放列表、显示总时长、显示播放进度等功能，满足了基本的播放需求，使用户拥有更原生态的体验^[2]。

【关键词】播放器；QT；Visual Studio；2017

1 引言

随着互联网的发展，娱乐是热门的话题，音乐作为最比较久远的娱乐项目，其载体播放器就显得极为重要。在互联网的发展过程中，涌现了各式各样的音乐播放器，这些播放器功能越来越丰富，给用户带来良好体验感的同时，也带来了诸多的问题，界面过于繁杂，广告越来越多。基于 QT 设计的 MP3 音乐播放器，给用户提供了友好的界面^[3]。

2 主要功能描述

本播放器基于 QT 进行界面设计，其功能实现上主要采用了 QT 封装好的 QMediaPlayer 类和 QMediaPlaylist 类进行实现。主要功能有：打开文件添加播放文件到本播放器，播放音乐文件，暂停和继续播放文件，停止播放，播放上一首、下一首功能，设置音量，设置播放的模式，播放列表，显示当前播放文件的总时长和播放进度功能。

3 播放器界面设计

3.1 总体设计

本播放器仿照了当前主流的播放器进界面设计，我们将 MP3 音乐播放器分为两个部分，一个是音乐播放器的首页部分；一个是播放器的播放列表部分。其中播放列表部分嵌套在播放器首页上，使用时点击按钮打开，不使用时点击按钮关闭，方便简洁。

3.2 播放器首页设计

在播放器的首页，主要设计了功能按键模块和显示模块。功能按键主要有播放按键、停止按键、停止播放按键、上一首、下一首、播放模式、音量设置、打开文件窗口以及打开播放列表按键。显示模块主要有显示当

前播放的文件名模块，显示当前播放文件的总时长，显示当前播放位置和播放进度条。

3.3 播放列表页面设计

播放列表页面的设计主要分为三个模块，第一个是列表模块，在这里显示加入的 MP3 文件的歌曲名和文件地址；第二个模块是信息显示，在这里显示了窗口标题，以及当前列表的歌曲总数；第三个模块是清空按钮，在这里进行了清空列表的设计。

4 功能实现

4.1 打开文件功能

QT 中的文件对话框 QFileDialog 类，根据 QT 官方文档给出的资料，QFileDialog 类最简单的使用方式就是调用静态函数来创建对话框获取文件，官方文档一共给出了八个静态函数。在本播放器项目中，主要使用到了 getOpenFileName 这个静态函数，例如：`QString pathFile = QFileDialog::getOpenFileName(this, "OpenFiles", "..\\", tr("MP3 files(*.mp3)"));` 用户可以用该代码段建立一个文件对话框，并且指定只能显示 MP3 类型的文件，选择后将获取选择文件的绝对路径，并且将改绝对路径保存到 pathFile 中。

但是这段代码存在一个问题：一次操作只能打开一个文件。在播放器的实际应用中，这样的设计就非常不方便。根据官方的说明文档，我们还可以得到另外一个静态函数 getOpenFileNames()，该静态函数可以获取多个本地文件，并将得到的文件路径保存到一个 QStringList 类型的变量中，例如：`QStringList pathFiles = QFileDialog::getOpenFileNames(this, "OpenFiles", "..\\", "MP3 files(*.mp3)");` 同样用户可以根据该代码段建立一个文件对话

框, 并且指定只能显示 MP3 类型的文件, 不过此时用户可以一次性选择一个或多个文件进行获取路径的操作。由于获取到的路径保存到了一个 QStringList 变量中, 而我们需要对一个一个路径进行操作, 因此我们需要用到迭代器来获取 QStringList 变量中的每一个值。

4.2 播放音乐功能

QT 中的 QMediaPlayer 类允许播放媒体源, 它可以用来播放歌曲、视频等内容。在 QT 给出的说明文档中, 指出了该类将播放内容指定为 QMediaContent 对象, 可以视为附加了附加信息的主 URL 或规范 URL。在 QMediaPlayer 类中, 给出了许多的枚举值, 例如播放状态、错误信息、媒体状态信息、播放标志位, 还给出了许多的属性, 我们可以用 get 和 set 来进行获取和设置。例如:

```
QMediaPlayer *mediaPlayer=new QMediaPlayer;
mediaPlayer->setMedia(QURL:: fromLocalFile
(pathFile));// 激活并设置媒体源
mediaPlayer->setVolume(50);// 设置媒体播放音量
mediaPlayer->play();// 播放
```

该代码段就可以实现简单的音乐播放功能, 但是仅仅只是调用 QMediaPlayer 类, 实现过于简单, 每次只能播放最后加载进来的文件, 就算可以实现列表的播放, 但是那样代码行比较多, 实现相对更复杂。

而 QMediaPlaylist 类就可以为 QMediaPlayer 提供一个播放列表, 这个列表实际上就是 QMediaContent 对象的列表。我们首先需要定义一个 QMediaPlaylist 类型的指针变量, 并且初始化为 NULL, 并且在打开文件获取文件路径的时候, 通过 addMedia 函数向列表添加多个媒体文件。QMediaPlayer 通过函数 setPlayList 来设置一个播放列表, 例如: mediaPlayer->setPlaylist(mediaPlaylist);

4.3 暂停、继续和停止播放功能

暂停和继续播放功能实现相对较简单, 本 MP3 音乐播放器将播放、暂停、继续三个功能集中放在了一个按钮上, 通过点击按钮, 改变按钮显示的文字来进行功能的选择。在进行功能的选择之前, 我们需要先对播放列表 QMediaPlaylist 进行是否为空的判断, 如果为空, 则弹出一个信息提示窗口提示当前播放列表为空; 如果不为空, 则先获取按钮上的文字, 如果等于“播放”, 则调用前面定义好的播放函数按照播放列表进行音乐文件的播放, 并且将按钮上的字设置为“暂停”; 如果等于“暂停”, 则使用 QMediaPlayer 中的 pause 函数将当前播放文件的播放状态设置为暂停, 并且将按钮的文字设置为“继续”; 如果等于“继续”, 又继续调用 play 函数继续之前的播放, 此时将按钮上的文字设置为“暂停”。

停止播放和暂停播放, 虽然都是将播放状态停止下

来, 但是区别在于暂停播放只是将播放暂时停止在当前状态, 而停止播放是将播放列表又回到开头, 重新点击播放时从开头开始播放。和暂停、继续相同, 停止播放也只需要调用 stop 函数, 并且将播放按钮上的文字设置为“播放”。

4.4 播放上 / 下一首功能

上一首和下一首按键的处理函数实现也相对简单, 只需要对播放列表 QMediaPlaylist 调用 previous 函数和 next 函数即可, 但是本 MP3 音乐播放器考虑到对当前播放文件的文件名显示, 对上一首按钮处理函数的实现是将当前播放文件索引设置为当前播放文件的上一个索引, 对下一首按钮处理函数也是同样如此, 实现如下:

```
mediaPlaylist->setCurrentIndex(mediaPlaylist-
>previousIndex());
mediaPlaylist->setCurrentIndex(mediaPlaylist-
>nextIndex());
```

上一首、下一首按键中除了对播放列表的改变以外, 还有对显示当前播放文件名字的内容的改变。在这里涉及到了 QT 中父窗口和子窗口的传值问题。当父窗口向子窗口传值, 在子窗口的 public 下定义一个函数用来获取父窗口的传值内容, 再在父窗口传值的位置调用子窗口中定义的函数。当改变了当前的播放文件索引之后, 获取当前播放文件的索引并保存到一个 QString 变量中, 通过调用播放列表窗口定义的函数将该索引传递到播放列表窗口中, 这样就可以通过这个索引来找到索引对应的文件名。

返回这个文件名的时候用到了子窗口向父窗口传递信息, 实现原理为: 在子窗口定义一个 signals 信号, 在需要向父窗口传递信息的位置用 emit 发射信号, 再在父窗口定义相应的 signals 信号, 并自定义一个槽函数接收该信号, 注意对信号和槽的连接。本播放器中, 当播放列表窗口找到相应的文件名的时候, 通过 emit 将文件名传递到主窗口中, 并在定义的槽函数中通过设置标签控件的文本方式调用了传递到主窗口的值, 显示了当前播放文件的文件名。

4.5 设置音量功能

设置音量主要包括两个部分, 一个设置音量大小的 slider 控件, 一个是显示隐藏 slider 控件的按钮。

首先是对音量调整控件的隐藏显示设置, 当本 MP3 音乐播放器启动时默认音量调整控件是隐藏的, 因此就需要在构造函数中调用一个隐藏窗口的函数 hide(), 在实现隐藏 / 显示音量控件按钮的处理函数的时候, 设置一个静态变量, 用来保存点击按钮的次数, 并通过 if 从句进行判断, 当点击次数为单数时打开音量设置控件,

当点击次数为双数时关闭音量设置控件。

然后是对音量调整控件的设置，因为我们通常情况下习惯的音量范围在 0 到 100 之间，所以我们先在构造函数中设置 slider 控件的最小值为 0，最大值为 100，同时设置默认状态下 slider 的值为 50。在调整音量函数中，调用 QMediaPlayer 中的 setVolume 函数来设置音量大小，并且将 slider 控件的值传递到 setVolume 函数中。这样，当我们滑动 slider 滑块的时候，音量也得到了相应的改变。

4.6 设置播放模式功能

在 QMediaPlaylist 中，包含有播放模式的枚举值：CurrentItemOnce 当前选中的媒体文件仅播放一次，CurrentItemInLoop 当前选中的媒体文件循环播放，Sequential 从当前选中的媒体文件开始，列表中的文件顺序播放一次直到最后一个文件，Loop 列表中的文件顺序循环播放，Random 列表中的文件随机播放。我们只需要在播放模式按键的处理函数中使用 if 条件从句和 QMediaPlaylist 中的 setPlaybackMode 函数便可实现播放模式的改变，例如：

```
QString text = ((QPushButton*)sender()->text();// 获取  
按键上的文字
```

```
    if(text== QString::fromLocal8Bit(“顺序播放”))  
    {  
        mediaPlayList->setPlaybackMode(QMedia  
aPlaylist::Loop);// 列表循环  
        ui.pushButton_playmode->setText(QStrin  
g::fromLocal8Bit(“列表循环”));  
    }  
}
```

这段代码片段首先获取了按键上的文件，当按键上的文字为“顺序播放”时，点击播放模式切换按钮，QMediaPlaylist 就通过 setPlaybackMode 函数将当前播放模式改变为列表循环，同时把按键上的文件设置为“列表循环”。其他的模仿模式实现同理。只需要将 setPlaybackMode 的参数设置为相应的枚举值即可。

4.7 播放列表功能

播放列表功能由于是单独的一个窗口，所以需要添加新的类。通过点击按钮显示该窗口，并将窗口用 move 函数固定到主窗口上。再次点击按钮，调用 hide 函数隐藏该窗口。

4.7.1 显示播放文件

在实现显示播放文件到列表的功能的时候，同样用到了前面说到的父窗口向子窗口传值的方法，这里不做赘述。显示文件的列表我们使用了一个 QTableView 控件，并且设置 QTableView 控件的加载模式，在构造函数中先

定义好列表的表头和风格，然后再在自定义的函数中将主窗口传递过来的值依次显示到列表中。

4.7.2 双击列表文件播放音乐功能

双击列表的时候，我们首先要获取获得焦点那一行数据的索引，在通过索引找到列表中文件的名称，将当前播放列表的文件索引设置为鼠标双击那一行的索引值，并将改行的文件名传回到主窗口。

4.8 显示总时长功能

要想显示总时长，想获取当前播放文件的总时长。在 QMediaPlayer 中，已经给出了获取当前播放文件总时长的函数 duration。但是当实际使用的时候，当直接调用该函数的时候，返回值无论如何都为 0。通过查阅帮助文档，找到了解决这个问题的办法。在初始化回放开始时可能不可用，连接 durationChanged 信号以接收状态通知，即需要自定义一个槽函数，为了与 duration 区分命名为 getduration，并在这个自定义的槽函数下面调用 duration 函数就可以接收到当前播放文件的总时长，注意在构造函数连接信号和槽的时候，信号为 durationChanged，槽函数为自己定义的 getduration。

Duration 获得的总时长是以 ms 为单位的，我们需要自己将毫秒转化为时：分：秒显示的格式，即可显示当前播放文件的总时长。

4.9 显示播放进度功能

显示播放进度分为两个部分，一个是显示当前播放位置的时间显示，一个是当前播放位置的进度条显示。

首先设置进度条的显示，我们想通过 QTimer 类型设置一个定时器，自定义一个槽函数，通过连接定时器设置 slider 控件随播放移动。首先定义一个全局的 QTimer 类型的指针变量，并定义一个全局的整型变量，用来设置进度条的最大值。并且在构造函数中将 slider 控件 enable 类型设置为 false，这样是为了 QMediaPlayer 在播放前禁止 slider 滑动。然后再自定义的槽函数中，将 enable 重新设置回 true，并且将 slider 控件的数据范围设置到 0 和之前定义的全局整型变量，同时在这里调用时间函数，为 slider 设置刷新的定时器，最后连接自定义的定时器槽函数。最后自定义一个定时器槽函数，函数的内容为将 slider 控件的显示刷新为当前播放的位置，这个值由当前播放位置和最大播放时间计算得到。

然后是当前播放时间的显示，当 slider 控件随播放时间移动设置好后，显示当前播放时间就更简单了。将自定义的显示当前时间的槽函数连接到 valueChanged 信号上面，这样自定义的函数就可以获得当前播放的位置，这个时间还是以毫秒为单位的，我们在使用的时候要把他转化为时分秒的形式显示。在这个函数中我们还可以

通过调用 QMediaPlayer 的 setPosition 函数来进行拖动滑块改变播放进度条。

5 结束语

虽然本 MP3 音乐播放器已经实现了音乐播放的绝大部分功能,但是站在产品的角度,还有许多值得优化的地方,比如进度条的显示还不够顺滑,可以增加歌词显示的功能。在以后的学习过程中还需要多加了解,仔细钻研。在努力提升自身知识和技术的同时,更加注重实

践的重要性。

【参考文献】

- [1] 鄢涛,刘永红,赵卫东,余悦,曾谊,于曦.基于 Qt 的高性能网络音乐播放器的设计与实现[J].成都大学学报(自然科学版),2017,36(01):55-58+69.
- [2] 刘晓立,赵俊逸.基于 Qt 的音乐播放器[J].软件导刊,2015,14(10):112-114.
- [3] 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.