

基于 QT 的音乐播放器的设计与实现

李鑫 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】随着社会的飞速发展,人们生活水平的日益提高。人们获取音乐的享受变得十分的容易。从前可能还需要买一个磁带、一个播放磁带的随身听。可是现在人们只需要打开电脑的一个软件,便可以播放自己想要的音乐,极大地丰富了人们的生活。本文以音乐播放器的应用为例,介绍怎样利用基于 QT 为开发框架设计和实现一个简洁的音乐播放器。通过基于 C++ 开发技术,利用 QT 的图形界面框架,就能够实现我们的需求。使用 QT 框架不仅具有体积小、播放格式多、占用资源少等优点,除此之外,它还具有跨平台的有点。它在其他平台只需要再编译一次,即可运行播放。极大地方便了用户。

【关键词】音乐播放器; QT; C++; QT 框架

引言

最近这几年随着计算机技术的飞速发展,电脑变成了每个人家里面不可缺少的一个电子产品。在日常生活中,人们除了办公之外,更多的是用来放松、娱乐。而听音乐便是一种人们能够轻易获得的娱乐方式。随着目前社会的各种计算机技术的百花齐放,涌现了各种各样的一些音乐软件。可是,这些音乐软件大多是广告繁多、功能复杂、占用系统内存。有些设计可能自己不是很喜欢。极大地让用户体验降低了一个档次。互联网上有非常丰富的音乐资源,也有不少基于这些资源的音乐播放器。这些音乐播放器普遍功能都较强大,但出于商业因素,通常会嵌入很多广告、新闻,甚至会经常出现弹窗给用户带来影响。同时,由于软件功能^[1]。所以,目前一款简洁、简单的音乐播放器,深受人们喜欢。

1 QT 技术的选择

再选择用 QT 还是 MFC 作为外部界面设计时,首先想到的 MFC。它的运行效率要比 QT 高,并且 MFC 的库要比 QT 更加齐全。

然而 QT 的某些方面可能会更加符合我的要求。QT 做的 GUI 开发要比 MFC 要好,并且 QT 界面库支持 CSS,界面设计更方便更美观。面向对象的特性体现的比 MFC 明显,在命名,继承,类的组织等方面保持了优秀的一致性,代码写起来比较优雅。QT 不仅可建立艺术级图形界面,更是一个跨平台的 C++ 图形用户界面应用程序框架^[2]。可在 Windows、Linux、Unix 等多平台开发。QT 是一个跨平台的 C++ 应用程序开发框架,它包括一套跨平台的类库、一套整合的开发工具和一个跨平台的

集成开发环境。利用 QT,可以有效地重用代码,使用一个代码库就可以适配多个主要桌面、嵌入式和移动平台。

2 主要功能描述

由于本应用程序主要是为了给用户一个简洁、方便的音乐播放器,所以客户端没有任何的广告,也没有任何的联网搜索功能。仅仅是把下载的歌曲能够播放。所以,大致包括的内容有:播放列表的排列、播放、暂停、上一首、下一首、音量、进度调节、打开本地列表。

用户首先同过点击打开本地列表的按钮把所要想播放的音乐添加到音乐播放列表中。然后点击播放按钮就可以播放音乐,也可以直接点击音乐播放。点击暂停就可以暂停播放。点击上一首则开始播放下一首音乐。点击下一首则开始播放下一首音乐。拖动音量按钮可以调节音乐声音的大小,也可以直接点击喇叭,使音乐静音。还可以拖动音乐进度条,使音乐播放到我们想要听的部分。

3 界面设计

本程序就只有一个界面,充分的体现了该程序的简洁的优点。该界面主要有音乐播放列表、上一首按钮、下一首按钮、播放按钮、暂停按钮、歌曲进度条、音量进度条、打开本地文件按钮。

页面顶端显示的是音乐播放器的专属 Logo。整个界面是一个宽 362、高 563 的矩形界面。整个界面是以一张图片为背景,整个图片可以选择自己喜欢的图片。可以使用户感到心情愉悦。以一张图片为背景的基础上,我们设计了一个透明的播放列表,可以展示用户喜欢的

图片。在播放列表的下面一排，有着歌曲的正在播放时间、歌曲的播放时间和正在播放歌曲的名字。再下一排是歌曲的进度条，可以拖动这个进度条来实现播放歌曲时间的变化。下一排也是最后一排，在整个界面的最下面。从左至右依次是上一首、播放\暂停、下一首、音量键图标、音量进度条、打开本地文件按钮。

4 功能实现

4.1 应用首页

首先，先按照里面功能的排版布局，大致的分析一下需要用多大的窗口才可以实现所要求实现的所有功能。所以，在大致排版布局之后的窗口大小是设置的宽362、高563的矩形界面。

关于这里的设置有两种方法。第一个是直接在里面设置，第二个是可以在QT的UI图形界面设置。

在UI图形界面中，首先打开属性编辑器。在里面的选项中找到 geometry 属性，在整个属性里面找到宽和高，分别填写362、563，整个窗口的大小就固定了。

代码里面可以直接用 setFixedSize()，这个方法可以直接设定窗口的固定大小。括号里面就可以填写宽度和高度。

代码示例：setFixedSize(362, 563);

除此之外，我们整个界面的左上角有个图标，这个是音乐播放器的LOGO。关于他的设置我们也还是有同样的两种方法。

在UI图形界面中，先打开对象查看器，查看选择的对象是否是最大的那个窗口。如果不是我们就选择最大的那个窗口。接着打开属性编辑器选择 windowIcon，点击后面的按钮选择合适、美观的一张图标，即可完成图标的设置。

代码里面可以直接使用 setWindowIcon()，括号里面填写路径。但是路径我们需要用 QIcon() 来包装一下，让 setWindowIcon() 能够识别。

代码示例：setWindowIcon(QIcon(“:/Resources/music.png”))

4.2 乐播放列表的设计和实现

音乐播放列表是用的QT组件中的 QListWidget。它相当于是一个列表，用以显示歌曲的基本信息。在UI图形界面中，选择在左边的 Widget Box 中找到 List Widget，并把它拖动到中间设置的固定的窗口中。按照之前规划好的布局，在这个窗口中拖动改变大小。返回到代码界面中，运行之后发现此时窗口之中的 List Widget 是一个白色窗口。挡住了我们设置的精美的图片。此时可以用 setStyleSheet() 设置 List Widget 的格式为

透明的。

代码示例：ui.listWidget->setStyleSheet(“background-color:transparent”);

接下来便是要实现如何添加歌曲到播放列表中。先创建一个函数 addfile()，再用 QMediaPlaylist 申请一个播放列表 *PlayerList。在 addfile() 中，首先先申请 QStringList 类型的 fileNames 存放从 QFileDialog 中取出来的文件的基本信息。写一个循环函数 i 从0开始，每次加一，直到大于或者等于 fileNames.size 时结束循环。在这个循环中播放列表 PleyerList 和 QFileInfo 类型的 info 在每次循环中要添加一个从 fileNames 存取的文件。接着，我们把 info.fileName 也就是文件的名字，通过调用 QListWidgetItem *item 把文件的文字传入 *item 中，再调用 addItem() 这个方法把 item 添加 ListWidget 中。在每次循环中添加一个歌曲名字，一直到大于或者等于 fileNames.size()。

调用 addfile() 函数时，需要用到 QT 的信号连接机制。之前设计界面时，有一个打开“本地文件按钮”，此时我们需要用到 connect() 来调用。这个 connect() 方法先写信号发送对象，接着对象发送的信号，接受对象，信号槽函数。这里的发送对象是本地文件按钮，发送的信号是 QPushButton 中的 clicked，接受对象就是这个窗口所以用 this，信号槽函数就是 addfile()。通过点击本地文件按钮，会弹出一个让我们选择文件的窗口，选择完之后，歌曲信息就会添加到播放列表中。

代码示例：connect(ui.file, &QPushButton::clicked, this, &MusicPlayer20::addFile);

4.3 播放、上一首、下一首

这三个功能总体来说十分简单，每个功能对应了一个函数，最后通过一个 connect() 把按钮和槽函数连接起来，就可以完成功能的实现。

播放功能的实现是在 PlayMusic() 中。首先，申请了 BOOL 类型的 ispaly 用来判断当前音乐是什么状态 (false 暂停、true 播放)，初始值默认 false。接着，我们还需要申请一个 QMediaPlayer 类型的 *Player 播放对象。在函数中选择了 if 和 else 来判断当前歌曲是什么状态。第一个判断 isplay==false，这时需要把 isplay 的状态置为 true 同时还需要把暂停按钮换成播放按钮，最后再把播放对象 Player 设置为 play() (播放) 状态。当判断为 ispaly==true 时，先把 isplay 状态置为 false，再把播放按钮置换成暂停按钮，最后把播放对象 Player 设置成 pause() (暂停) 状态。最后，我们通过 connect()，使按键连接槽函数。发送信号对象为播放\暂停按钮，信号使 clicked，接受对象是这个窗口 this，槽函数 PlayMusic，这样就实现了播放歌曲时点击就会实现暂

停播放图标变成暂停图标。

代码示例: connect(ui.play, &QPushButton::clicked, this, &MediaPlayer20::playMusic);

下一首和上一首的实现就很简单了。上一首是在 preMusic() 中实现。只需要在里面把播放列表 Playlist 指向上一首 previous() 歌曲。而下一首是在 nextMusic 中实现, 在这个函数中需要把播放列表 Playlist 指向下一首 next() 就可以了。然后用两个 connect() 分别把上一首按钮和下一首按钮连接起来, 就能够完成上一首和下一首的功能。

4.4 歌曲进度条的设置、音量大小的设置

歌曲进度条和音量的进度条都是 QT 中的 QSlider 中的横向进度条, 配合着 QT 中 multimedia 使用起来很方便。里面有特定的信号和槽函数可以设置进度条的大小。

首先, 关于歌曲进度条相关方面的设置这里有点复杂。滑动进度条要改变快进、和倒退音乐这里要用到两个函数: positionChange()、seekChange()。它们的主要功能分别是: 一个是使进度条获取当前正在播放歌曲的时间, 一个是使当滑动条数值改变时, 歌曲也在改变。在 positionChange() 中我们首先用播放对象 Player 通过 duration() 使歌曲进度条获取当前播放歌曲的整个时间。接着设置如果音乐进度改变, 则改变滑块位置。接着我们设置一个 int 类型的 move 存放音乐播放进度。用 move 分别除以 60000、1000(这里时间以毫秒为单位), 让时间以 00:00(前面是分后面是秒)的形式出现在进度条的上方。在 seekChange() 中, 首先让播放对象 Player 调用 setPosition()。效果是当滑动进度条改变时, 歌曲进度也在改变。

最后再通过两个 connect() 使按钮与槽函数连接起来歌曲进度的改变也就实现了。不过这里要注意的是第一个 connect() 中, 信号的发出者是播放对象 Player, 它发出的信号是 positionChanged, 对象是本窗口, 槽函数是 positionChange()。实现的是当歌曲的信息传入时我们可以使进度条获取歌曲的时间信息。第二个 connect() 中对象是歌曲的进度条发出的信号是 sliderMoved, 接收信号的本窗口, 槽函数是 seekChange()。这样当我们滑动进度条时, 歌曲的进度也会随之而改变。

关于音量的大小的设置, 这里就显得很简单了。这里直接用了 connect(), 直接在里面的槽函数写。这里的槽函数需要用到 lambda 函数。这里信号的发出者是音量进度条, 信号是 QSlider 中的 valueChanged(), 接受信号对象可以省略不写了。接着我们直接再 lambda 中先让播放对象 Player 能够设置音量大小 setVolume()。随后我们用判断语句来判断音量状态从而可以改变音量图标。当音量进度条 value==0 时, 图标改成静音标志。当 value != 0 时, 图标设置成正常音量图标。

除此之外, 还可以通过直接点击音量图标改变音量大小。这里也是直接再 connect() 中写, 这里槽函数还是 lambda()。对象是音量按钮, 信号是点击, 接受信号对象可以省略。槽函数里面直接用判断语句就能实现这两个功能。这里需要申请一个 bool 类型的 sound 来判断音量的状态(true: 静音、false: 非静音)。当 sound==true 时, 把音量进度条大小设置为 0; 图标改成静音图标, 并把 sound 状态改成 false。当 sound==false 时, 把音量进度条大小设为一个任意默认值。图标换成正常音量图标, 并把 sound 状态改成 true。到此, 音量进度条相关的所有功能都实现了。

5 结束语

本文介绍了如何使用 QT 基于 multimedia 实现了音乐播放的简单功能。适应 QT 框架的应用具有跨平台、占用资源少等特点。并且 QT 的代码是基于 C++ 语言, 是大众普遍学习过的语言。并且 QT 这几年越来越流行, 并且里面的 UI 设计界面也十分方便, 设计出来的 UI 效果也很出色。相比较与其他音乐播放器, 本应用还是有很多的不足之处, 但是里面的内容通俗易懂, 非常适合 QT 有关媒体播放的初学者学习。

【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.