基于 QT 的音乐播放器的设计与实现

唐海涛 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】目前文娱行业发展迅速，使得人们的空闲时间也丰富多彩。享受一首美妙的歌曲是目前快节奏生活中的一种调味剂，给人带来轻松愉悦的享受。本文以 QT 为例，如何利用 QMediaPlayer 框架实现引入 MP3 文件、进度条控制音乐当前播放进度及音量大小、播放上一首下一首、切换播放模式等功能。利用 QT 优良的跨平台特性，以及它提供给开发者建立艺术级图形用户界面所需功能的特性，设计了一款可 移植的音乐播放器 [1]。

【关键词】音乐播放器；QT；QMediaPlayer

引言

随着社会的进步与发展，人们开始不止满足于物质生活，而更加注重于精神生活。在如今的高压 力的生活、学习、工作中，音乐是放松精神，增加生活趣味的不二选择 [2]。近年来人们越来越追求生活品质，文娱行业也蓬勃发展。在平时的生活中，听歌的需求也变得不可或缺，因为歌曲可以给人带来精神上的放松和享受，也可以带动人的情绪，比如近年来非常火的各大短视频 APP，就是简单地音乐加视频的方式。而想要在 PC 端欣赏歌曲就需要一款用户心仪的音乐播放器。相较于市面上已有的成熟商用的播放器，自己设计开发的播放器又有可以更符合自己的实际需求和审美，也无需看插入的广告等优点。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，由于软件功能的多样化和复杂性，这些音乐播放器很难做到性能优异 [3-4]。

1 开发框架的选择

在考虑选择使用什么框架来开发 PC 端的音乐播放器时，首先想到的就是 QT 和 MFC。MFC 和 QT 都可以做出界面，QT 的优势在于，QT 做出的界面更精细，更精美，更专业，可以满足用户对于界面美观的绝大部分需求，而 MFC 虽然也能做界面，但是入门的门槛比 QT 高，需要掌握 MFC 的相关的一些基础的接口，而且使用起来也没有 QT 方便，QT 更易入门，接口简单，最重要的是，QT 做出的项目，支持跨平台，可移植性高，所以使用 QT 在 Windows 操作系统下开发的音乐播放器，在 Linux 下也可以完美运行，而不用开发者去修改任何的代码，很大程度上降低了开发时兼容平台测试时所需的时间和精力，减少了开发成本。

2 主要功能描述

音乐播放器，顾名思义，需要能够从 PC 文件中获取 MP3 文件的路径，即打开文件，并且能实现播放该文件，能够切换歌曲，实现上一首，下一首的按键功能。用户还能根据自己的需要，随时可以暂停该 MP3 文件，或者切换到自己喜欢的播放模式如随机播放，列表循环，还能通过可视化的进度条控制音乐播放的进度和音量大小。也可以根据自身审美，选择不同的窗口背景。

3 界面设计

根据市面上已有的音乐播放器和用户需求的简单易操作性，我们将音乐播放器分为一个界面，可以看到播放，暂停，上一首，下一首，打开，和播放模式的下拉框。常用的基本窗体控件有 Label 标签控件、 TextBox 文本框控件、 Button 按钮控件、 Panel 面板控件等等。 这些控件 的组合排列就可以构成不同的界面效果 [5]。

4 功能实现

4.1 读入打开文件

读入文件是通过用户按键的方式，当用户鼠标点击打开文件 BUTTON 时，就会发出一个信号，并执行相应的槽函数。所以定义一个名为 openfile() 的槽函数，先引入头文件 QFileDialog 和 QFileInfo，定义一个 QString 类型变量，用来存储 MP3 文件的路径，使用 QFileDialog 中的方法 getOpenFileName()，用处是弹出一个对话框，由用户选择一个文件，当用户选择了文件后，返回该文件的绝对路径，由于是音乐播放器，所以我们可以设置

该方法的第四个参数为”(*.mp3)”，就只会看到 mp3 文件了，如果有其他类型的音频文件，也可以用同样方法进行设置。获取到选中文件的绝对路径后，使用 Playlist 中的 addMedia 方法，将路径作为参数传入，该文件就被添加到了 Playlist 播放列表中了，但还并不会显示到用户可视的窗口上。所以需要使用某种方法，获取到该文件的文件名，并将其设置到窗口上的 QTableWidget 控件上。获取文件名的方法为 QFileInfo 中的 setFile() 方法。此时需要定义一个 int 类型变量 index，用于获取当前 QTableWidget 中的行数，故当多次打开文件时，使用 insertRow 方法，以 index 为参数，能够正确插入到对应行。最后需要连接信号和槽函数，使用 connect，连接点击 BUTTON 时发出的信号和 openfile 这个槽函数。点击 BUTTON 发出的信号为 QPushButton：：clicked。

4.2 播放暂停实现

同样是采用用户按键的方式，当用户点击播放 BUTTON 时，发出一个信号，执行对应槽函数。定义一个名为 playFile() 的槽函数。槽函数内容为先判断播放列表是否为空，为空就不执行任何操作，不为空就播放当前指针指向的列表中的文件。判断是否为空的语句为 if(Player->mediaStatus ()!= QMediaPlayer::NoMedia)，播放文件用的方法是 Player->play()。暂停和播放是同样的实现，定义一个名为 pauseFile() 的槽函数，同样先判断播放列表是否为空，不为空就暂停当前指针指向的列表中的文件。暂停文件使用的方法是 Player->pause()。最后分别使用 btn_play 和 btn_pause 连接两个槽函数，使用的信号也是 QPushButton::clicked。

4.3 上一首下一首实现

定义两个槽函数，nextFile() 和 previousFile()。先进行一个判断，判断当前的播放模式，是否为单曲循环模式，如果是，则无论点击上一首或下一首，都依然是播放当前的文件。判断语句是 if(Playlist->playbackMode() == MediaPlaylist::CurrentItemInLoop)，在判断体内定义一个 int 类型变量 currentIndex，将 Playlist 指向的当前文件索引赋值给 currentIndex，再用 Playlist 中的 setCurrentIndex() 方法，设置指针指向的索引，参数为前文定义的 currentIndex，最后利用 Player 中方法 play() 播放文件。若不是单曲循环模式，再进行一次判断，是否为随机播放模式，判断语句为 else if(Playlist->playbackMode() != QMediaPlaylist::Random)，如果不是，则就是正常的播放上一首下一首，即列表循环或者顺序播放。首先是上一首，在 previousFile() 函数体内先定义一个变量 currentIndex，将 Playlist 当前指向的索引值赋值给 currentIndex 变量，记录下当前索引，再做一次判断，若当前索引值为零，则表示已经在播放列表首部，此时再点击上一首，应该将播放列表的指针指向播放列表的尾部，即最后一首歌曲，判断语句为 if(--currentIndex == -1)，控制指针指向的位置的语句为 currentIndex=Playlist->mediaCount()-1。同理下一首操作时，先判断当前索引值加一是否等于播放列表文件数，因为索引是从零开始的，如果相等，则代表当前指针指向的位置为播放列表的尾部，此时点击下一首，应该设置指针指向播放列表的头部，判断语句为 if(++currentIndex == Playlist->mediaCount())，控制指针指向位置语句为 currentIndex=0。回到第一层判断，此时只剩下一种播放模式，即随机播放模式，当用户点击上一首下一首执行的操作应该都是相同的，即在零到播放列表文件数之间生成一个随机数，该随机数不等于当前指针指向的索引值，再将指针指向的位置重新设置，最后进行 play() 操作即可。生成随机数的方法为 int currentIndex = qrand() % (Playlist->mediaCount))。最后分别连接槽函数，方法为 connect(ui->btn_next, &QPushButton::clicked, [=](){nextFile();})。上一首的连接同上。

4.4 切换播放模式实现

切换播放模式使用的控件和前面的 QPushButton 不同，使用的是下拉框，QComboBox。该控件的下拉菜单，每一行都有一个索引值，我们可以根据这个索引值来获取到用户选择的是什么播放模式。同样，先定义一个名为 playbackmodechanged() 的槽函数，槽函数内定义一个名为 index 的 int 类型变量，用来记录 QComboBox 控件的当前索引值，通过 switch 语句，分别重新设置当前播放模式，设置播放模式的方法为 Playlist->setPlaybackMode(QMediaPlaylist::PlaybackMode::xxx)，可以通过 Qt 帮助文档具体查看播放模式的枚举值。最后连接槽函数，连接方法为 connect(ui->comboBox, SIGNAL(currentIndexChanged(int)), this, SLOT(playbackmodechanged()))。

4.5 进度条控制音乐播放进度及音量实现

使用的控件为 QSlider，要想实现音乐播放时，进度条动，需要将进度条控件，取名为 slider_Progress 和文件播放发出的数值改变信号相连接，连接的方法是 connect(Player, &QMediaPlayer::positionChanged,ui->slider_Progress,&QSlider::setValue)，连接后只能实现音乐文件播放时，数值改变，可以发出信号，进而改变进度条，但实现不了拖动进度条，而改变当前播放进度，此时还需要一个连接，上一个连接是播放的文件发出信号，进度条接收到信号，执行对应改变数值的槽函数，而改变了，此时就需要进度条能发出数值改变的信号，

播放的音乐文件能接收到这个信号，并执行相应改变数值的槽函数，连接方法是 connect(ui->slider_Progress, spSlider, Player, &QMediaPlayer::setPosition)。控制音量的进度条同理，取名为 slider_Volume，同样也需要双向连接，才可实现相互控制，连接同理。但在连接这两个槽函数之前，需要先定义两个函数指针，原因是 Qt 中 5.4 版本以上，connect 检查会比之前的严格，要求信号和槽函数必须满足一定的要求，才可执行，即信号中的参数必须大于等于槽函数中参数，切参数位置必须对应，所以在面临重载问题时，会比低版本检查更严格，定义函数指针的目的在于告诉编译器，用到的信号，参数只有 int 一个，定义函数指针的方法为 void(QSlider:: *spSlider)(int) = &QSlider::valueChanged。最后设置默认音量为五十，ui->sliderVolume->setValue(50)。

4.6 其他功能实现

使用 setWindowTitle( ) 方法设置窗口的标题。定义一个 QPalette 对象，使用其中的方法 pal.setBrush(QPalette::Backgroud, QBrush(QPixmap(图片路径))) 设置窗口背景图片。

5 结束语

本次实现的音乐播放器可以实现打开文件，播放暂停，上一首下一首，换肤，进度条控制播放进度和音量等功能，但是提升空间仍然还很大，站在商用角度还有很大的改善空间。比如能否实现歌词的播放，能否实现网络音乐播放器，评论功能等。在今后的学习中，应该多注重界面设计，即用户的交互性及体验。

【参考文献】

[1] 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.
[2] 吕尚伟 . 随身听与随身看 [J]. 数码影像时代 ,2012(12):2.
[3]Blanchette J,Summerfield M.C ++ GUI Qt 4 编程 [M]. 闫锋欣，曾泉人 , 张志强 . 北京 : 电子工业出版社 ,2013.
[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(7):148-150.
[5] 杨岫 , 朱露露 , 李志国 , 何宗林 . 基于 Windows 系统的音乐播放器设计 [J]. 电脑编程技巧与维护 ,2017(03):77-78.