

# 基于 QT 的视频播放器设计与实现

包秋英 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,大量的视频播放带动了视频播放器行业的发展。本文讲解的是基于 QT 的视频播放器,开发环境为 windows 平台下的 QT,QT 是跨平台的 C++ 图形用户界面应用程序开发的框架<sup>[1]</sup>。

**【关键词】**视频播放器; QT; C++

## 1 背景

在这个飞速发展的社会里,看一些视频已经成为了人们的消遣,因此,视频播放器已经成为了人们生活中必不可少的一部分。视频的播放也成为了视频播放器的产生必要的条件。而大部分的视频播放器在播放视频的时候会有广告的插入、打开视频速度慢以及视频卡顿等缺点,从而给用户可以说是带来一些不良的视频观看的体验。

## 2 图形界面的选择

MFC 是专门为 windows 设计的,它是一个用于开发图形用户界面的类库<sup>[2]</sup>。MFC 使用了面向对象的方法包装了 Win32 的 API,也是因为这样,这些 API 有时是 C,有时是 C++,也可以是 C 和 C++ 的混合体。

QT 是跨平台的,语言结构简单清晰,QT 良好封装机制使 QT 的模块化程度很高,可重用性比较好,对于用户开发来说,QT 是非常方便的,面向服务对象的特征可以体现得比 MFC 明显;MFC 较 QT 的代码比较杂乱和乱,写 MFC 必须首先要知道 Windows 的消息机制。

## 3 主要功能描述

视频播放器主要实现了对本地视频的播放、对本地视频的停止以及对本地视频的暂停等功能,并且点击播放、暂停以及停止以后可以看到当前的播放状态;可以通过拖动进度条的位置从而改变视频的播放进度,视频的播放进度可以在设置的静态文本框中看到,如视频在当前的播放进度,和视频的总持续时间;可以拖动进度条的位置来改变视频的饱和度、亮度以及对比度、色相以及音量;可通过直接打开文件或者打开目录的方式来播放视频,以打开目录的方式打开视频时,需要鼠标双击视频列表中的某一个视频文件才能够播放该视频,并且单击时不能编辑此视频名称。

## 4 功能实现

### 4.1 按钮功能实现

#### 4.1.1 环境配置

在 QT 中想要实现视频播放的功能,就得先在文件的 .pro 文件中加入代码 `QT+= core gui multimedia multimediawidgets` 代表在项目中导入该模块,该模块提供了多媒体的展示和出来,还提供了一些访问录音机和摄像头的必要 API<sup>[3]</sup>。

在实现播放按钮前,需要先定义两个新类 `QMediaPlayer` 和 `QVedioWidget`。类 `QMediaPlayer` 是可以从源文件进行播放媒体,可以很方便的播放指定的文件,且提供了很多设置和获取信息的接口;类 `QVedioWidget` 是一个用来播放视频的控件,可以将类 `QVedioWidget` 理解是 `QMediaPlayer` 的一个输出端口。定义完一个新类以后,设置类 `QVedioWidget` 对象 `vedioWidget` 的输出框大小,使用 `setVideoOutput` 函数给类 `QMediaPlayer` 的对象 `player` 设置视频输出框到 `QVedioWidget` 对象 `vedioWidget`,并且使用 `setMedia` 设置默认播放的视频的来源。以上操作均在初始化函数中进行。

#### 4.1.2 播放按钮实现

在完成以上环境配置以后,播放按钮的实现就显得特别简单,直接给类 `QMediaPlayer` 对象 `player` 用函数 `play()` 就可以实现视频的播放。但为了使按钮看起来美观,还设置了按钮的样式表,其中,包括按钮 `button` 的背景颜色,按钮的形状以及按钮在鼠标按下时的下沉样式。

#### 4.1.3 暂停按钮的实现

直接给类 `QMediaPlayer` 对象 `player` 用函数 `pause()` 就可以实现视频的暂停。但为了使按钮看起来美观,暂停按钮的样式与播放按钮的样式设置相同。

#### 4.1.4 停止按钮的实现

直接给类 `QMediaPlayer` 对象 `player` 用函数 `stop()`

就可以实现视频的停止。但为了使按钮看起来美观, 设置了按钮的样式表, 按钮的样式与播放按钮的样式设置相同。

#### 4.1.5 打开文件按钮的实现

在打开按钮的槽函数中引入一个新类 `QFileDialog`, 并将其引用写入头文件中, 运用该类中 `getOpenFileName (NULL, "标题", ".", ".WMV")` 函数, 该函数括号里的参数分别为: 指定的父类、标题、默认打开文件后显示的目录、右下角的文件过滤器(在此以 WMV 为例), 并用 `setMedia` 函数播放此视频。但为了使按钮看起来美观, 打开视频按钮的样式与播放按钮的样式设置相同。

#### 4.1.6 打开目录按钮的实现

在点击打开按钮是应该先将文件名称对应的列表和文件对应的路径以及对应的数据模型先清空, 并且在打开目录的时候不能播放开始播放的视频。使用 QT 中 `QFileDialog::getExistingDirectory` 的方法打开一个已经存在的文件夹并将文件夹里的内容放在目录类里边, 在将文件夹里的内容放入目录类里是, 还需要设置一个筛选器, 将视频的格式进行筛选, 比如需要 `wmv` 格式或者是 `mp4` 或者其他格式。在打开文件目录是, 需要将打开的文件进行计数, 所以定义了一个 `count` 进行文件的计数, 使用 `appendRow` 函数将获取的文件名称加入到行里边, 然后将打开的文件名称写到 `QTableView` 界面视图组件中。在此设计的是在打开文件目录以后, 需双击列表才能够播放视频, 所以给 `QTableView` 界面视图组件设置了一个双击的槽函数进行播放, 并且双击不能编辑文件, 所以使用 `setEditTriggers` 函数实现。但为了使按钮看起来美观, 打开目录按钮的样式与播放按钮的样式设置相同。

### 4.2 滑块功能实现

#### 4.2.1 亮度滑块功能实现

在亮度滑块的实现中, 为了美观, 先给亮度滑块设置了样式表, 其中设置了滑块的颜色、样式、背景颜色、滑块划过的颜色以及鼠标拖动的样式等等, 可以通过拖动滑块的位置来进行视频的亮度调节。具体实现是给滑块通过 `setValue->brightness()` 函数表明是对亮度进行调节, 再使用 `connect` 函数将滑块的改变值与视频数据进行同步以达到亮度调节的功能。

#### 4.2.2 对比度滑块功能实现

在对比度滑块的实现中, 为了美观, 先给对比度滑块设置了样式表, 其中设置了滑块的颜色、样式、背景颜色、滑块划过的颜色以及鼠标拖动的样式等等, 可以通过拖动对比度滑块的位置来进行视频的对比度调节。具体实现是给滑块通过 `setValue->contrast()` 函数表明是对对比度进行调节, 再使用 `connect` 函数将滑块的改变值与视频数据进行同步以达到对比度调节的功能。

#### 4.2.3 饱和度滑块功能实现

在饱和度滑块的实现中, 为了美观, 先给饱和度滑块设置了样式表, 其中设置了滑块的颜色、样式、背景颜色、滑块划过的颜色以及鼠标拖动的样式等等, 可以通过拖动饱和度滑块的位置来进行视频的亮度调节。具体实现是给滑块通过 `setValue->saturation()` 函数表明是对饱和度进行调节, 再使用 `connect` 函数将滑块的改变值与视频数据进行同步以达到饱和度调节的功能。

#### 4.2.4 色相滑块功能实现

在色相滑块的实现中, 为了美观, 先给色相滑块设置了样式表, 其中设置了滑块的颜色、样式、背景颜色、滑块划过的颜色以及鼠标拖动的样式等等, 可以通过拖动色相滑块的位置来进行视频的色相调节。具体实现是给滑块通过 `setValue->hue()` 函数表明是对色相进行调节, 再使用 `connect` 函数将滑块的改变值与视频数据进行同步以达到色相调节的功能。

#### 4.2.5 音量滑块功能实现

在音量滑块的实现中, 为了美观, 先给色相滑块设置了样式表, 其中设置了滑块的颜色、样式、背景颜色、滑块划过的颜色以及鼠标拖动的样式等等, 然后使用槽函数 `sliderMoved()`, 表示当用户拖动滑块, 发射信号。其中音量的调节使用 `setVolume` 函数。

#### 4.2.6 视频时间滑块功能实现

在准备视频滑块的功能前, 需要知道的是播放的视频时长为多少, 所以设置了两个静态文本框用来记录视频的总时长和当前播放的进度。首先, 需要一个改变时间格式的函数, 需要把秒转换成时分秒的格式, 在这个函数里边, 分钟是总秒数取整 60, 秒钟是总秒数取余 60; 当分钟大于 60 时, 小时是由分钟取整 60 而来, 分钟由分钟取余 60 而来。然后将得来的时分秒进行格式化, 如果时分秒数值小于 9 的时候, 就在时分秒数值前加 0, 如果时分秒的数值大于 9, 就直接用当前的数值。最后输出时分秒的格式。

在静态文本框设置当前的播放进度, 需要使用改变时间格式的函数, 然后使用 `setValue` 函数设置滑块的位置。在静态文本框设置视频的总时长, 也需要使用改变时间格式的函数, 其中视频的最大时长直接用 `QMediaPlayer` 类中的 `duration` 函数, 因为视频的最大时长需要的是秒为单位, 所以可以用 `duration/1000` 来表示秒。

接下来就是视频播放滑块的设计, 在视频播放滑块中也设置了滑块的样式表以达到美观的效果。视频的播放需要知道当前的进度和总进度, 所以使用 `setRange` 函数获取滑块范围, 并使用 `setPosition` 函数获取滑块调整位置, 从而达到拖动滑块改变视频进度的效果, 调整位

置以后需要更新滑块并且更新时间。

## 5 结语

本文介绍了在QT环境下视频播放器的设计与实现，且代码简单易懂，界面简洁统一。虽然这次实现的视频播放器能够实现视频正常的播放、暂停、停止功能，在视频的输出框中也能实现对视频的播放进度、音量、亮度、饱和度、色相、对比度的调节，但是还是可以进行更多的完善，本视频播放器现在只实现了播放本地视频，

后边还可以实现网络视频播放器的些许功能。

## 【参考文献】

- [1] 赵北庚. Windows 系统下 Qt 图形界面集成开发环境搭建 [J]. 科技经济导刊, 2016 (09): 20.
- [2] 朱光洪. 基于 MFC 的视频播放器的设计与实现 [D]. 湖南: 湖南大学, 2017.
- [3] 陈云鹤. 基于 Qt 的嵌入式媒体播放器的设计与实现 [D]. 湖北: 华中科技大学, 2006.