

基于 QT 的视频播放器设计与实现

杨明子 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】视频已经成为了我们生命中不可或缺的一部分了，无论是抖音、快手这种短视频还是优酷、腾讯视频、爱奇艺这种大型的视频软件，他们的主要目的都是播放视频，一个简简单单的视频播放，在程序员的眼中可能并不是那么完美，本文针对 QT 视频的设计制作进行简单的分析，以供参考。

【关键词】视频播放器；QT；C++

1 系统简介

该视频播放器是在普通的视频播放下进行的功能完善与扩充，也就是说，除了视频播放功能，还设计了其他个性化的功能如：亮度的调整、对比度的调整、色相的调整以及饱和度的调整，当然也有视频播放的基本功能：播放、暂停、停止、打开文件和将文件夹的内容添加至文件列表并双击打开的功能。

2 系统界面设计

2.1 总体设计

在我的调研中，整理出来了许多客户心中的视频播放器所应该具备的内容，我们可以将软件分为以下 3 个部分：播放区、优化区和列表区。实现的功能分别为为基础的视频播放、视频的个性化调整和将文件加入到列表中。我希望该软件可以让大家方便的、个性化的、批量化的播放视频。

2.2 播放区的设计

播放区内是由 3 个 button：播放、暂停和停止。点击播放按钮时会播放视频，暂停时视频将会暂停等待下一步操作，点击停止时视频将会回到原点。在 3 个 button 的上面还有一个进度条 slider 是用于标明视频播放进度的部件，进度条的下放会有两个随着时间推移而改变的文本框，第一个是用于表现视频播放进度的，第二个是用于表现是视频本身长度的。为了美观，这两个文本框的值都是通过 QString 类型并以 **: **: ** 返回的。

2.3 优化区的设计

优化区的优化功能主要体现在 4 个方面：亮度、饱和度、对比度和色相。优化区是体现本项目“个性化”的地方，为了让用户自己选择视频的滤镜和格式，我们用到了 4 个进度条，每一个进度条都是对这四个方

面的调整，他们相互独立互不干扰，可以结合调试以达到最好的观看体验。

2.4 列表区的设计

列表区主要实现的是批量化，如果用户需要将观看的内容一个个的导入，不仅会在查找视频上显得繁琐，也会使用户的观看不连续。所以我采用了列表区，列表区的主要组成部分有三个插件，有两个 button（“打开文件”、“打开目录”）和一个 Listview，打开文件是单个打开视频文件，打开的文件会加入到 Listview 中。当使用打开目录按钮时，选中的不是单个文件而是文件夹，选中文件夹后会将文件夹中所有的视频文件加入到列表中。当列表中有文件后就可以双击点文件名，这样就可以打开视频进行播放。

3 功能实现

3.1 本例开发环境

该视频播放器是使用的 QT creator5.9.0 作为的开发环境，所使用的计算机系统为 Windows10，也可以在 Windows 8 以上的系统使用。起初是想用 VS 2015 进行编译，但是想锻炼自己的 QT 上的开发技巧，就决定在 QT creator 上完成该项目，VS 2015 也是可以进行 QT 的编译，但是 VS 2015 中没有通过 ui 中的部件进行“转到槽”的功能（当然也可以手动输入代码），但是为了代码的准确性和可读性，我选择在 QTcreator 上完成该项目。QT 是一个跨平台的 C 图形用户界面库，由挪威 Troll Tech 公司出品，目前包括 QT，基于 Framebuffer 的 QT Embedded，快速开发工具 QT Designer，国际化工具 QT Linguist 等^[1]。QT 还有许多其他的功能除了对界面的设计，还有基于 Qt 的 Windows 下指令调用及标准输出的监控获取、人机交互界面的设计、数据的过滤与分析、检查结果的显示及保存^[2]，本项目主要应用的库文件是

<QMediaPlayer> 的库，引用这个库文件必须要在 .pro 文件中的 QT += 后加入 multimedia 和 multimediacore 两个语句。

因为会在该项目中播放不同的本地视频，所以首先需要将视频放入到本地文件夹中（最好放在同一个文件夹下），以便查找和添加多条视频进入系统。因为会用到 QT 的 DEBUG 所有要加入 <QDebug> 的头文件，在后期的文件列表中需要获取文件名列表等功能需要加入 QT 中的文件管理相应的头文件：<QDir>、<QStandardItemModel>、<QFileDialog>、<QFileInfo> 还有播放视频需要用到的 <QVideoWidget>。

3.2 播放区的实现

播放、暂停、停止应该是最简单的一个部分，分别点开三个 button 进行“转到槽”操作，并使用 <QVideoWidget> 库中包含的 player->play() 进行播放，player->pause() 进行暂停，player->stop() 进行停止。在播放区最难的也是本项目最难的地方之一就是进度条的编写。进度条是反映视频播放进度的所以要将其设定为用户不可改变的，再者进度条一定是和时间相关的，需要定义一个 position 来返回当前时间的值，但是返回到的所有时间都为秒，所以在设计滑块的变动前首先要设计了一个类 changeTime（返回值为 QString），这个类用于将秒转化为小时和分钟并以 **: **: ** 的形式返回（例：00: 00: 00）。分钟为 position/60，秒为 position%60，但是当得到的值小于 9 时就需要将值改为两位数就需要在前面加上一个 0 这里用到了 QString::min = (min>9) ? QString::number(min) : tr("0") + QString::number(min); 做一个判定在这里可以返回，这样就可以完成视频进度的时间化，返回为 **: **: ** 是为了增加精确度和可读性。最后需要设计一个类 updatePosition，因为在设计滑块时，滑块并不会随着视频的播放而更新滑块的进度，所以设计了这个类。这里用到了一个变量 duration 这个量是来自 <QVideoWidget> 库的是用于返回视频的总长度，duration/1000 就是视频的总长度的百分比在这里用到的是 setMaximum 还有一个对应视频播放的值就是 position，将 position/1000 就可以得到此处滑块的进度，这里用的是 setValue，每隔一定的时间，滑块就会更新一次，就像是计算机进程工作原理一样，宏观上是一起运行的其实微观上是独立运行的，计算机中有很多这样的例子，滑块的移动也是，用户看来是一个滑块一直运动，但是再程序员的实现其实是几个时态的滑块不停地更新。这样就是播放区的全部实现了，播放区的实现主要依赖于 <QVideoWidget> 库中设计好的类与方法。

3.3 优化区的实现

优化区中含有的四个要素分别是：亮度、对比度、

饱和度和色相。优化区主要是为了实现个性化，个性化就需要用户自己的调节，所以为这四个要素都分别设置了 4 个滑块，分别控制这四个要素，用户可以通过这四个滑块的组合调整成自己想要的效果。在这个区域中最主要的还是 <QVideoWidget> 因为在这个库中，有四个对应的方法：亮度 (brightness)、对比度 (contrast)、色相 (hue) 和饱和度 (saturation)。为了实现滑块对于视频效果的调整加入了以下的两条语句：connect(ui->horizontalSlider_2, &QSlider::sliderMoved, videoWidget, &QVideoWidget::setBrightness);

```
connect(videoWidget, &QVideoWidget::brightnessChanged, ui->horizontalSlider_2, &QSlider::value);
```

顾名思义就是将用户改变的对比度和滑块进行连接，使用户很方便的就可以对视频效果进行调整。这就是本项目最简单的一部分：优化区的实现。

3.4 列表区的实现

列表区中含有 2 个按键，一个是用于单独添加文件的按键，另外一个为添加整个文件夹所有视频的按钮。在这个区域，设计了两个成员变量：strListFileName 和 strListUrlName，前者是为了使文件名称对应列表，后者是为了文件对应路劲列表，在我们的列表区的 Listview 区域中，我们看到的应该是所属的文件名而不是路径名称，所以就会涉及到文件名和文件路径的映射。

首先是打开文件，这里就会用到几个库文件：<QDir>、<QStandardItemModel>、<QFileDialog>、<QFileInfo>，首先设置一个变量 fileName 用于显示文件名称 QFileDialog::getOpenFileName() 用于打开固定格式的文件。

然后是打开目录的设计，在每次打开目录前都会做一次清除，将文件路径、文件名字、数据模型和视频都会做一次 clear 操作，然后设置一个 dirname 用于设置文件夹名字，使得后面可以直接通过文件夹完成导入，QString dirname = QFileDialog::getExistingDirectory(this, tr("open Directory"), "."); 这条语句用于打开文件夹，还需要获取文件名列表、目录的路径和文件的个数，这里文件的个数其实是可以不定的，后期可以通过一个 for 循环，循环中用 strListUrlName.append 的方法添加文件个数，循环中还要加入 append->appendRow，使得列表中会出现这些文件，以便后来的栓剂列表即可打开文件。

最后就是刚刚说的双击列表打开文件，首先在“转到槽”操作时要用到 DoubleClicked 的方法转入，然后需要 setMedia（路径选中的那一行）然后进行播放操作 (player->play()); 这就是本项目的最后一个部分列表区的设计了。

4 结束语

虽然这个项目可以基本的完成视频播放，但是还有很多不完美的地方，比如不能播放网上在线的视频，也不能对所有的视频有记忆的存储播放进度，将所有的视频格式都可以打开并播放，今后对于计算机的学习将不再是代码本身，而是产品的需求性和健壮性，要思考如

何才能做一个容错率高的软件供大家使用。

【参考文献】

- [1] 陈琦 .QT 的编程技术及应用 [J]. 科技信息 .2008(33).
- [2] 王星 , 郭皓蓉 . 基于 Qt 对 Windows 自动安全检查的研究与应用 [J]. 信息与电脑 (理论版),2020,32(05).