

基于 Qt 实现的播放器项目

郝瑜柯 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】目前互联网科技技术的高速发展,各种自主研发的互联网项目层出不穷。在此设计并实现一款适合自己的播放器,不仅提高了自身技术的发展,带给自己舒适满意的体验感。本文基于 Qt 编程环境的基础上,结合 C++ 语言编程知识,介绍如何使用 Qt 实现一款满足自己心意的播放器项目。

【关键词】Qt; 编程; C++; 播放器

1 引言

近些年来,互联网技术的逐渐完善与发展,人们生活水平的逐渐提高与完善,越来越多的娱乐项目被广大的互联网用户所接受。至此,通过自身学习的编程技术,设计与实现一款属于自己设计的界面视频播放器,并编程实现,不仅总结了自身的知识技能,也开拓了眼见。

2 开发环境的选择

本项目使用的是基于 Qt 实现的播放器,为何选择基于 Qt 编程实现,而不是基于 MFC 的界面实现呢。第一,出于自身实力,考虑到 MFC 的界面设计难度要大于 Qt 针对于新手而已;第二,功能丰富,Qt 除了提供界面库^[1]之外,还提供了音频库,3d 库,数据库 SDK, WebEngine, 网络库等,让程序员在做这种跨平台的程序开发时,变得简便。总之 Qt 有着编译环境自带的音频库,会使得编程更加轻松,所以首先想要完成这个项目就必须要在自己的电脑上安装 Qt5.0 以上版本的软件,并配置好相对应的 Qtcreator 环境。

第三,Qt 也不依赖硬件环境。一个操作系统下开发的应用,放到另一个操作系统下依然可以运行。针对每一种操作系统平台,Qt 都有一套对应的底层类库,而接口是完全一致的。因此只要是在 Qt 库上^[2]开发的程序,放在任何一种平台下都可以编译运行,也就是说在操作系统和应用层之间,增加了一个平台层来保证可移植性。总体而言,相对于 MFC 的界面设计,本项目的设计与实现更适合使用于基于 Qt 的编程环境实现。

3 主要功能描述

本应用产品的使用定位,是面向想做出满足自己心意界面,实现自身编程实现的播放器,所以在此基础上,需要实现的功能并不复杂。并不需要涉及到数据库等后

台程序功能的连接。简单来说项目的组成被划分为三大区域,视频播放的界面显示,视频待播放区列表的显示,一系列操作视频的按钮,简称为功能区。

通过点击,在视频待播放区列表里面的打开视频文件与添加播放按钮,实现对视频的添加操作。并将在视频播放区列表中点击的视频,显示在视频播放的界面上,完成用户交互,实现视频播放。紧接着就是功能按钮区的实现,例如在此项目中有视频进度条的拉取,视频的开始按钮,视频的暂停按钮,视频播放的倍速快慢,与结束视频播放的响应按钮,调节亮度的高低滑块,甚至于音量大小的调节等等。

4 项目界面设计

4.1 总体设计

本项目总体来说,将界面的整体划分为三个部分组成,分别对应的是,第一点:视频的显示播放窗口,第二点:视频目录的列表窗口,以及最后,在控制视频播放的功能上,实现一系列功能按钮的组合。通过 Qt 的槽函数事件绑定与响应,达到与使用者,实时人机交互的方式。三个部分的规划上,视频显示播放部分是最好事先的,会首先介绍。其次是视频添加目录的列表部分,最后是控制视频播放的功能实现按钮的组合设计。

4.2 视频界面设计

在播放视频的界面上,我们只需要使用一个界面控件,从 Qt 的 ui 界面中拖取 verticalLayout 的布局方式,放在基类的初试界面上,并且在 .cpp 文件中,通过代码编写,将视频添加到布局去,这样就实现了视频界面的设计。

4.3 列表窗口界面设计

在视频目录列表的显示上,首先设计一个 groupBox

装入所有的需要添加在此部分的子控件，并留出从顶端出发三分之一的空白部分，加入，打开视频文件按钮与添加播放列表按钮，以及占据三分之二部分的 listView 显示窗口。

4.4 功能界面设计

通过 Qt 的 ui 界面直接拖取 Qslider 控件，与 QPushButton 控件，结合 label 标记的备注，实现以开发者想要的设计方式进行排布界面设计，没有固定的位置区域选择。真正实现适合自己，让自己满意的编程。

5 功能实现

5.1 视频播放

在考虑要如何将播放的视频显示到界面上时，强大的 Qt 已经为我们解决了这一切的烦恼，在 Qt 的库中，为了开发者提供了一个叫做 QMediaPlayer 的媒体播放类和 QVideoWidget 的视频显示类，两者的组合，可以用于解析音频文件和视频播放的显示，直接继承于 QMediaObject。首先，就是要定义一个叫做 player 的 QMediaPlayer 和 VideoWidget 的变量对象，将声明的 VideoWidget 使用 addWidget 函数操作方式添加到 verticalLayout 中，通过 ui 中拖动 verticalLayout 大小，实现视频播放界面的大小，再把 player 通过 setVideoOutput 函数添加到 VideoWidget 变量中，至此，就能实现了一个的播放视频的窗口。

5.2 列表添加

在列表的显示方面，界面显示上需要将添加的视频播放文件名对齐，所以我们需要先声明一个叫做 QStandardItemModel 的数据类模型。QStandardItemModel 的数据类模型是基于每一项数据的标准数据模型，可以处理二维数据。其中每个项都是一个 QStandardItem 类的变量，用于存储项的数据、字体格式、对齐方式等，有多个行和多个列，每个基本显示单元是一个单元格。

在添加文件目录窗口中，首先要考虑的是在界面上的显示，而显示的内容是，文件的名称列表。紧接着就是考虑文件播放时的问题，实质上播放的内容还是在电脑上存储的视频文件，所以这里我们需要添加每一个视频文件所对应的路径的列表。为了达到与使用者操作简单的方式，我们这里就需要为打开视频文件按钮与添加播放列表按钮添加信号槽函数，两者的功能在前面已经介绍，这里简答说明所使用的函数实现，当鼠标点击文件打开按钮时，使用 getOpenFileName 函数，获取打开文件的默认路径操作，并点击按钮的信号，从而实现打开视频文件按钮的实现。

所以在此基础上，添加列表的 listView 中代码中，

通过循环的操作方式，使列表目录的显示更加快捷。定义 strListFileName 变量和 strListUrlName 变量，分别显示文件名和文件路径。紧接着通过当前路径，使用文件操作的方式获取目录文件路径转换成 QString 类型，添加到变量 dir 中，再根据文件的后缀名，使用文件筛选器选中符合条件的视频播放的信息。最后在文件列表中要点击单个视频文件名称，实现播放，就需要实现双击打开的功能，此功能的实现，需要在 ui 的控件中定义信号槽的双击事件响应，通过索引传递的方式，选中当前的行文件，使用 QUrl 的 fromLocalFile 将 player 的路径改为当前文件路径进行播放。

5.3 功能区按钮

在功能区的按钮中，第一，最容易实现的三组按钮分别是播放，暂停和结束按钮。这些按钮的功能通过 ui 界面的信号槽函数的点击函数进行声明，在 cpp 文件中将播放和暂停按钮的函数实现进行写入，设置为 play 函数，pause 函数与 stop 函数。

第二，是倍速模式的播放按钮，通过不断点击倍速按钮，会有播放速度的显示，此功能的实现是基于 Qt 音频库中的 setplaybackrate 函数，结合 switch+case 选择相对应的点击次数，每点击一次就实现一次播放速率的累加，但在鼠标点击次数超过 4 次时，又需要重新设置速率，置其为 1，进而实现播放速率的循环改变，最终形成播放按钮的实现。

第三，视频亮度的调节方面，不难发现有许多的 public function，例如亮度 brightness，对比度 contrast，饱和度 saturation。所以借助在 QVideoWidget 中已有的亮度函数 brightness 实现亮度调节。由于亮度是一个可变的范围值，所以，我们需要通过在 ui 界面定义的 slider 控件，写两个槽函数的 connect 实现，首先将 QVideoWidget 的 brightness 移动到的滑块数值传递给 sliderMoved，再通过另一个槽函数连接视频亮度与 sliderMoved 的值进行连接，从而达到获取数值，滑块移动，视频亮度三者之间的相呼应，滑块值的改变相对应的视频亮度值也随之改变。

第四，是音量的大小设置。音量的大小调节也使用 slider 类的方式实现，但由于 QVideoWidget 中没有提供公共的方法，所以需要手动实现，将槽函数的定义设置为 valueChaged，通过 player 的 setVolume 的方式获取 horizontalSlider->value 的值，相应的设置到信号槽的连接上。

第五，视频进度条的改变，相应的视频播放的进度也随之发生改变。通过分析，得知首先需要得到视频播放源总的播放时间 duration，将其设置为 slider 滑块的滑动范围大小，再者，声明并定义 positionChange 函数，

将滑块移动的位置所在的数值传递给 slider。继续通过声明 setPosition 函数,在内部实现 player 位置的定义改变。最后,将总的视频播放时间和所在滑块所在的播放位置,两者通过槽函数分别连接到 QMediaPlayer 上,实现滑块的拖动,到达视频状态的变化。

6 结束语

本文介绍了如何使用 Qt 一款跨平台 C++ 图形界面应用程序的开发框架,制作播放器的教学原理。项目结

构单一,内容简洁,还存在许多可以优化的地方,但尽管如此,本项目的开发时间成本低,很适合学习 Qt 的新手开发。

【参考文献】

- [1] 黄渊,吴磊.基于 Qt 面向对象程序设计[J].湖南农机,2011,38(05):59-61.
- [2] 徐艳菲,吴铁峰.基于 Android 的音视频播放器的研究与设计[J].微处理机,2017,38(06):74-77.