

基于 QT 的视频播放器的设计与实现

韦梦婷 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】现今市场上的视频播放器各有千秋，但又都有不尽人意之处，比如需要占用大量内存、广告多以及某种格式的视频只能用相应的播放器播放等。本文设计并开发的这款视频播放器，以简洁为特点，旨在让使用者可以免去各种广告、捆绑软件、占用大量内存等困扰，给使用者一个最简洁最原始的使用体验。

【关键词】视频播放器；QT；C++

引言

在互联网技术极速发展的当下，人们接触到了各式各样的视频播放器，但过于商业化的视频播放器总有各种各样的广告或限制，虽然功能繁多，但在某些特定情况下对于某些用户来说反倒是一种负担，比如只是简单的需要打开某个视频时，突然发现想要打开这个视频还得下载某个软件，众多的操作和下载捆绑就会让用户十分困扰。本文设计并开发的这款视频播放器就可以解决这些问题，给用户一个十分简洁明了的播放器。

1 开发环境介绍

QT 是诺基亚开发的一个跨平台的 C++ 图形用户界面应用程序框架。它提供给应用程序开发者建立艺术级的图形用户界面所需的所用功能。QT 是完全面向对象的，很容易扩展，并且允许真正地组件编程^[1]。

2 主要功能描述

本视频播放器主要分三大模块的功能，一为基础部分，二为列表部分，三为调节部分。基础部分主要包括播放、暂停以及停止播放，列表部分主要包括打开文件和打开目录，调节部分主要包括音量、亮度、对比度、色相以及饱和度调节。用户可以通过打开目录将本地文件夹中的视频批量导入到列表中，并以文件名显示在列表中，通过直接点击列表中的文件名打开对应视频进行播放。同时，用户也可以通过鼠标拖拽进度条调整播放进度，音量、亮度、对比度、色相以及饱和度也可以通过鼠标拖拽来改变。

3 界面设计

应用界面十分简单明了，界面整体左上方为播放区域，左下方为调节区域，右方整体为列表部分。播放框

下面左边显示当前播放时间，中间显示播放状态，右边显示视频总时长。界面左下方调节区域分别有音量、对比度、色相、饱和度以及亮度的调节条。界面右上方设置了两个按钮，一个为“打开文件”，一个为“打开目录”，下方为列表显示框，显示打开目录下的视频文件列表。

4 功能实现

4.1 界面控件

进度条使用 Horizontal Slider 控件并命名为 horizontalSlider_progressBar；播放按钮使用 Push Button 控件并命名为 pushButton_play；暂停按钮使用 Push Button 控件并命名为 pushButton_suspend；停止按钮使用 Push Button 控件并命名为 pushButton_stop；亮度调节使用 Horizontal Slider 控件并命名为 horizontalSlider_brightness；对比度使用 Horizontal Slider 控件并命名为 horizontalSlider_contrast；色相使用 Horizontal Slider 控件并命名为 horizontalSlider_hue；饱和度使用 Horizontal Slider 控件并命名为 horizontalSlider_saturation；打开文件按钮使用 Push Button 控件并命名为 pushButton_openFile；打开目录按钮使用 Push Button 控件并命名为 pushButton_openDirectory；文件列表使用 List View 控件并命名为 listView_fileList；音量使用 Horizontal Slider 控件并命名为 horizontalSlider_volume；使用 label 控件设置播放当前时间、当前播放状态和视频总时长等，并分别命名为 label、label_1、label_2。

4.2 播放功能

作为一个视频播放器，最基本的需求就是播放暂停以及停止的功能，如何让用户有一个最简单的操作体验是本应用最大的追求，鉴于很多播放器设置的是鼠标点击播放界面就会将视频暂停，这样的功能看似方便，但对于一些用户来说是非常容易误点的，而误点对于很多用户来说也是一个很苦恼的事情，所以，本播放器就

将播放暂停和停止的功能以按钮的方式呈现出来, 用户想要将视频暂停只能通过点击播放框下面的暂停按钮, 而不会因为误点到播放框就暂停了。那么如何实现点击按钮就播放视频或暂停视频呢, 首先想到的就是通过设置按钮的 clicked 函数来响应用户操作, 再通过 QMediaPlayer 自带的方法 pause()、stop() 和 play() 来控制当前播放视频的播放状态, 当播放状态改变时使用 label 标签的 setText() 函数将标签内容改变为当前播放状态, 给用户一个最直观的展示。

由于在播放视频时经常需要调整播放进度, 比如直接跳过某个部分直接观看后面的内容或者倒回去观看前面的内容, 所以进度条以及进度条的拖拽也是必不可少的功能。首先需要让进度条与视频播放的进度相对应。用 setMaximum() 设置进度条的最大值, 将视频总时长经过一定换算后的结果设置为进度条的最大值, 随着视频的播放, 进度条根据获取当前播放时间并换算结果不停移动滑块, 进而实现进度条功能, 在用户视角上感觉到的就是进度条展示着当前播放进度。为了让用户可以对进度条进行调整, 使用 on_horizontalSlider_sliderMoved() 函数根据进度条滑块的位置变化来改变播放进度, 并通过 setText() 函数将标签内容改变当前播放时间。实现视频播放进度、当前播放时间标签与进度条三者同步。

4.3 调节功能

在观看视频时, 亮度、对比度、色相、饱和度和音量也是非常重要的调节功能, 尤其亮度和音量。首先将音量条的最大值和最小值分别设为 100 和 0, 初始值为 50, 通过滑块的上下位置移动信号与槽函数连接, 使用 setVolume() 函数将视频播放音量设置为音量条滑块对应的值。亮度、对比度、色相和饱和度的最大值和最小值分别设为 100 和 -100, 初始值为中间值 0 (也就是默认为视频原有的对比度、亮度、饱和度和色相)。利用 setValue() 函数分别把视频的亮度、色相、对比度和饱和度分别设置为亮度条、色相条、对比度条和饱和度条的值 (即滑块的位置), 并通过连接函数将视频播放框的亮度、对比度、色相和饱和度与滑块的移动信号关联起来, 通过连接函数分别将视频播放框的亮度、对比度、色相和饱和度设置为对应控件上滑块对应的值, 在用户视角上感觉到的就是通过滑动滑块就能改变视频的对比度、亮度、色相、音量和饱和度, 十分简单明了。

4.4 打开文件和目录

由于本播放器是一个本地视频播放器, 必不可少的功能自然是打开本地视频文件, 考虑到用户导入本地的某个文件夹里所有视频的需求, 设置了打开目录按钮, 可以将本地文件夹中的视频批量导入到列表中, 直

接通过点击列表中的视频名 (文件名) 播放指定视频, 只需要播放某一个单独的视频时, 就可以使用打开文件按钮, 这样播放器就只会播放这一个单独被选中的视频。

在打开文件按钮的点击响应函数中, 先设置一个 QString 类型的变量来接收使用 getOpenFileName() 函数获取到的需要打开的文件名, 再通过 setMedia() 函数将选中的视频放到播放框中播放出来, 为了避免选中非视频文件打开造成播放器崩溃, 使用 getOpenFileName() 时, 通过文件后缀名筛选出视频文件, 这样用户在打开文件的窗口中的文件夹里就不会看到非视频文件, 从而避免了非视频文件造成的播放器崩溃。

对于打开目录功能, 首先需要考虑到的是到底是打开的本地文件夹中的视频还是播放器中的视频, 为了减少播放器所占内存, 本应用打开的是本地文件夹中的视频, 将文件夹中的文件名导入到播放器的播放列表中, 通过点击文件名打开本地文件夹中对应的视频。那么首先需要考虑的就是设置一个存放路径的列表类型变量和一个存放文件名的列表类型变量, 通过文件名打开对应路径的文件并打开。在打开目录按钮的点击响应函数中, 设置一个 QString 类型的变量 dirname 来接收使用 getExistingDirectory() 函数获取到的需要打开的目录名, 定义一个 QDir 类型的变量 dir (目录列), 将获取到的目录名 dirname 放入目录列, 定义一个 QStringList 类型的变量 filter 来存放需要筛选的视频格式, 将自己想要的视频格式都放进去, 在打开目录时直接用 filter 设置筛选文件格式。通过 entryList() 函数将目录列中的文件名获取到文件名列表 strListFileName 中, 并通过 absolutePath() 函数获取目录的路径名, 最后通过循环语句将文件名列表中文件名一行一行地导入到模型中, 并将路径名存入到路径列表 strListUrlName 中, 最后把模型中的内容在播放列表中以文件名的形式显示出来。

除了设置打开文件和打开目录的功能外, 还需要设置播放列表双击响应函数, 在播放列表的双击响应函数中通过双击的视频获取路径名, 并在播放框中将对应路径的视频播放出来。

从用户视角上感觉到的就是点击本地目录后直接将文件夹中的视频导入到了播放列表中, 可以通过点击播放列表中的视频名 (即文件名) 播放视频。

5 测试及结论

项目完成之后, 对本视频播放器采用黑盒测试的方式进行了功能性测试。

点击打开文件按钮后, 选中本地文件夹中需要播放的某一个 mp4 视频, 该视频正常播放, 结论: 该功能正常。

点击暂停按钮后, 视频随之暂停播放, 并且进度条

保持当前进度不变,当前播放时间静止,播放状态标签变为“播放状态”,结论:该功能正常。

点击播放按钮后,视频随之继续播放,并且进度条继续随着播放进度改变,当前播放时间同步增加,播放状态标签变为“暂停播放!”,结论:该功能正常。

点击停止按钮,视频随之停止播放,并且进度条停止前进,当前播放时间保持不变,播放状态标签变为“停止播放!”,结论:该功能正常。

分别前后拖动进度条后,视频正常跳转,结论:该功能正常。

分别拖动音量条、亮度条、色相条、饱和度条和对比度条滑块后,音量、亮度、色相、饱和度 and 对比度随之改变,该功能正常。

点击打开目录按钮后,选中本地的某一个文件夹,该文件夹中的所有后缀名为 .mp4 和 .avi 的视频文件均以

文件名的形式呈现在播放列表中,但由于播放列表宽度不够长,显示不完整,结论:该功能正常。

双击播放列表中的某一个文件后,该文件在播放框中正常播放,结论:该功能正常。

总结:该视频播放器基础功能均能正常使用。

6 结束语

本文介绍了如何使用QT开发一款本地视频播放器,该视频播放器具有所占内存小,功能简单,界面简洁的特点,但相对于多数视频播放器还存在许多不足。由于开发成本低,代码简单易懂,非常适用于QT初学者学习练习。

【参考文献】

- [1]朱浩悦,张红祥.基于QT的视频媒体播放器的设计与实现[J].科技信息,2011(30):274-275.