

MFC 视频播放器的设计实现

邓建云 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】为了打造属于自己的专属视频播放器同时方便自己，本文基于 MFC+ 第三方库 vlc 打造一个属于自己的简单又方便的视频播放器不考虑联网和连接数据库等问题，实现了视频播放，MV 播放等功能。

【关键词】视频播放器； MFC； Visual c++； vlc

1 背景

现在市面上的播放器大多都是网络播放器，在播放本地视频的时候还需要下载相应的视频播放器才能播放本地视频，为了方便自己平时播放音乐或者视频，利用 MFC 我们可以自己制作一个专属自己的简易视频播放器，风格样式由自己来决定。

2 开发环境

开发工具是 Visual c++ 主要 c/c++ 语言的控制台程序开发，所以我们得选择一个界面工具来完成对界面的开发，到这里我们两种选择选取界面工具来完成我们的播放器，首先看下 qt 是现在做界面最常用的一种工具，而且是跨平台的，不过做起来比较麻烦，我们这是做自己的播放器就在 cp 端使用就不用考虑跨平台和联网等功能那些的功能所以 qt 可以排除。

这里主要选用比较方便快捷的 MFC 来制作界面，MFC 的话虽然局限于 cp 端，但是我们主要的就是实现在 cp 打造一个自己播放器使用 MFC 正好，由于 MFC 制作视频播放器不像 qt 那样可以不要第三方解码库，所以我们要选用一个开源视频编码解码库，vlc 是一种开源并且功能比较完善的视频解码库并且封装性比较好，跨平台，简单，这样涉及到的知识就包含了 MFC 上播放 gif 动画，MFC+vlc 库操作等。配置最简单的吧 vlc 文件和项目文件处在同一目录下在添加时不要用绝对路径最好是相对路径。

3 功能内容

本播放器的功能包含市面上的播放器的大部分功能，按键实现的播放视频、暂停、继续、拖动条实现的音量加减、进度拖动^[1]等其中暂停/播放由一个按键来完成，还有不可少的播放窗口，播放列表窗口，还有一个就是拖动视频到播放器的时候自动播放的功能和在拖拽视频文件进入播放器的同时，我们也将视频路径添加

到播放列表中，当拖拽多个视频文件时就可以选择点击播放播放视频或者是在列表中双击选中的视频进行正常播放。

4 界面及内容实现

4.1 界面布局

首先是一个播放窗口设置属性 Accept files 为 true 接收文件，这样在我们拖拽视频文件进入播放器的时候播放器才能做出相应的反应，占整个应用窗口的 75%，背景颜色我们可以设置为黑色，添加一个消息 WM_ERASEBKGDND 处理程序，使用 GetClientRect 来获得客户区的大小使用 FillSolidRect () 来填充底色。而我们播放位置的获取就是用 vlc 的 libvlc_media_player_get_time() 这个接口来完成，把它封装在一个类 GetCurrentPos() 中。进度条处于播放窗口的下方长多不超过播放窗口的宽度加个定时器 WM_TIME 处理程序 SetTimer () 方法，设置多长时间刷新一次然后使用 GetCurrentPos() 来获取视频播放的位置就是视频播放到的位置占整个视频长度的多少，然后使用 m_slider.SetPos () 设置控件的位置视频播放的时候进度条就会随着播放的进度改变进度条的位置，在这之前我们创建一个类来获得视频的长度，调用这个接口 libvlc_media_get_duration () 得到视频长度之后并返回，在进度条的下方我们可以选择一个 slider 空间来做为音量调节器，注意这里使用 mblieVLC 的 mediaPlayer.audio.volume 会发生失败要使用系统调节音量的方式来做才行，在获取到 slider 的值之后就可以通过 slider.value 改变音量的大小了，最下方就是播放暂停按钮这里我们可以点击按键的属性修改一下 icon 的值这样开做按键的美化，而在右侧则是把视频拖进来之后显示的播放列表。还有一点就是应用界面大小可以拉伸但是播放界面和播放列表不动，所以就要固定一下界面使用 Windows 的 API GetWindowLong 来获取参数固定对话框；~WS_MAXIMIZEBOX 去掉最大化窗口 ~ WS_MAXIMIZE

去掉最大边框 `~WS_SIZEBOX` 然后使用 `GetWindowLon()` 设置界面。

4.2 播放功能的实现

我们平常最常用的播放视频的方式都是直接把视频拖拽进播放器这样来播放视频, 首先添加响应函数 `WM_DROPFILES` 处理程序, 然后使用 Windows 的 API `DragQueryFile()` 来获取拖拽视频文件的路径, 当然如果同时拖拽的文件不止一个我们也可以用这个接口 `DragQueryFile()` 来进行计数然后再做一个循环来选择首先响应那个文件, 而 MFC 做视频播放器就需要第三方库, 这里选用 `vlc` 前面已经解释过 `vlc` 的优点, 引进工程完成配置之后创建一个播放类, 在创建三个成员 `libvlc_instance_t*`; `libvlc_media_player_t*`; `libvlc_media_t*`; `vlc` 的接口创建三个实例来准备播放, 顺序是播放实例、播放器(媒体), 还有必不可少的初始化函数播放前首先要给分配内存, 并且判断有无播放的实例, 播放函数 `player(char* vdi0)`, 判断是否存在路径之后 `libvlc_media_new_path()` 接口给内存分配路径创建项目, 其次 `libvlc_media_player_new_from_media()` 接口创建播放环境并且保存下来, 然后 `libvlc_media_release()` 释放刚才分配的内存以免造成常发性内存泄漏^[2], 最重要的来了, 使用 `libvlc_media_player_play()` 接口就可以直接播放了, 但是这样的播放是重新创建的一个独立的播放窗口, 所以还得划定一个区域给他让他在这个指定的区域内播放, 实现这个功能使用 `libvlc_media_player_set_hwnd()` 接口调用句柄选择在我们想要他在指定的地方播放, 在这个过程中会发现报错, 在这里就一定要把响应的动态库和插件复制到生成的 `exe` 文件同目录之下。

4.3 暂停 / 播放功能的实现

我们首先设置默认没有播放这里设置一个布尔变量来表达; 使用 `libvlc_media_player_set_pause()` 接口创建暂停函数 `pause()` 实现暂停的功能, 为了暂停之后能够继续播放这里需要获取视频的播放状态, 获取状态的方法有两个, 一个是用 `vlc` 的一个接口或者我们自定义一个变量表示, 这里简单一点的选择用一个布尔变量初始化为 `false` 表示没有播放视频, 当我们拖拽视频文件到播放器内的时候就改变变量值之后进行播放, 当需要暂停的时候添加一个判断当前视频的状态是否是播放的状态如果是播放状态就调用暂停函数 `pause()` 暂停播放并且改变变量状态。但是这里会有一个 bug 如果没有视频在播放器中的时候点击暂停 / 播放按钮的话播放器就会崩掉, 所以还得再添加一个判断是否有视频在播放才能保证应用不会死机。

4.4 文件目录解码

播放器完成之后有些视频拖进去能播放有些则不能, 发生这样的原因就是转码的问题, 一般情况下我们

要获取文件路径的时候包含中文的就播放不了这就需要在进行一下转码, 这调用 Windows 的一些 API 就能完成, `MultiByteToWideChar()` `WideCharToMultiByte()` 接口将输入的路径转为宽字符串之后再转为 `utf-8` 字符集这样就实现了对中文的路径的输入以及正常显示。

4.5 播放列表的实现

整个应用界面的右侧或者在自己喜欢的位置添加列表框 `list control`, 添加变量然后使用 `InsertItem()` 把路径添加进列表里, 然后使用 `GetFirstSelectedItemPosition()` 来获取鼠标点击的第一个信息, 然后使用 `GetNextSelecteditem()` 来定义选中的长度然后做一个判断, 为了友好的用户体验还可以加一个提示语, 在调用播放函数这样就可以。还有一种播放功能也是我们平时习惯性的操作, 在播放列表里双击我们希望播放的文件就能正常播放, 首先创建添加一个类向导先选中列表添加 `NM_DBLCLK` 处理程序再获取视频路径调用播放函数就完成了播放。

4.6 大图标以及小图标的设置

这里设置图标的文件是 `ico` 文件, 可以自己 `ps` 一张或者下载一张, 更换图标的方法有两个先说简单的, 最简单的就是找到资源文件的 `ico` 文件所在的目录把原系统存在的 `ico` 文件删除换上自己的图标重命名为原图标的名字重新生成一下代码就行了; 还有一种方法比较麻烦, 但是当腰添加多个资源文件时比较好用, 首先还是把自己用的图标放在 `res` 文件夹里在解决方案管理器哪里添加资源选择 `icon` 导入刚才添加的 `ico` 文件, 然后修改 `m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME)` 为 `m_hIcon = AfxGetApp()->LoadIcon(IDR_ICON1)`; 就实现了应用程序图标的修改, 但是 `exe` 文件的图标还没有被修改到就需要到 `resource` 头文件中修改 `icon1` 的值小于 `MAINFRAME` 的值这样 `exe` 文件的图标也就随之修改。

5 结束语

本文介绍如何实现打造一个属于自己的视频播放器, 实现了市面上一些视频播放器的主要功能, 满足了日常看视频的需求, 使用的 `c++ MFC + vlc` 视频编码解码库来开发的代码简单易懂能够快速上手, 学习的成本也是很低的, 就需要一点时间而已, 当然也还有不足的地方, 没有实现流式播放连接网络等功能, 这都是后面需要改进的地方。

【参考文献】

- [1] 陈琳. 变速视频播放器的设计与实现 [J]. 电脑知识与技术, 2009, 5(18): 4975-4976+4984.
- [2] 张天良. C/C++ 中常见内存泄漏与对策及预防措施浅析 [J]. 科技资讯, 2008(35): 8-9.