

基于 MFC 技术的视频播放器的设计与实现

杜巍 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】随着电子技术计算机技术的发展,多媒体技术的运用与更迭,在与计算机相关的技术发展中,各种音频视频的播放器随处可见,在人们的日常生活中,播放器几乎不可缺少,于是各种播放器紧跟着发展起来(如:腾讯视频,爱奇艺视频,优酷视频,QQ 影音播放器,暴风影音播放器等等)。笔者的这篇介绍是通过 MFC 技术和 VLC 库开发一个简单的视频播放器的过程,了解播放器的功能实现,以及播放器开发的的部分的相关技术,同时利用了软件工程的相关理论知识,了解软件开发的方法与步骤。该播放器是以 Windows 系统为平台,C++ 语言为开发语言,在 VS2017 界面下,进行设计与实现的。

【关键词】视频播放器; C++; VS2017; MFC; VLC

引言

随着计算机的普及应用,计算机应用软件得到了快速的发展,而且这种现象还会继续的维持,所以从某种意义上来说,计算机之所以能够在各个领域中得到应用,很大程度上就是因为相应的应用软件,然后根据分析各个行业的特点,开发了针对性的应用软件,客户和通过这些软件的使用,能够给实际的工作带来方便,提升工作的效率,生产技术得到了保证即人们的基本物质需求得到了基本意义上满足,那么接下来就是人们对精神需求的探索。其中视屏(包括电视剧、电影、综艺,甚至是现在很火的短视频)都是人们消磨时间,感受文化,愉悦充实自我的途径,那视频播放器的需求就得到了增加。但是笔者自己能力有限,缺少开发经验,此次设计肯定存在许多不足之处,随着我知识的积累和经验的增加,笔者将会逐步对它完善,使整个系统从操作简便、界面友好、灵活、实用、功能全面的要求出发,体现该播放器的稳定性,拓展性,兼容性的特点。

1 简介

1.1 播放器举例简介

当前小节只是进行举例,目前比较使用较多的播放器,比如 QQ 影音播放器、暴风影音播放器等。

1.1.1 QQ 影音播放器

1.1.2 暴风影音播放器

1.1.3 风行影音播放器

2 软件开发及开发工具

2.1 软件开发平台(Windows)

Microsoft Windows 使我们现在普遍使用的操作系统,

其特点在于直观、高效的面向对象的图形用户界面,易学易用,并且由于它的操作简便画面质感等原因逐渐变为众所周知的操作系统。

2.2 C++ 语言

它是我们大学中最常见的一门编程语言,也是我们软件工程专业同学一进校就开始学习的一门基础语言 C 语言的集合,C++ 语言的使用范围特别的广泛,全球的编译语言中也名列前茅。它支持过程化程序设计、数据抽象、面向对象程序设计、制作图标等等泛型程序设计等多种程序设计风格。

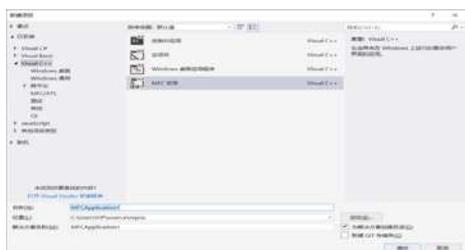
2.3 VLC

VLC 多媒体播放器(最初命名为 VideoLAN 客户端)是 VideoLAN 计划的多媒体播放器。它支持众多音频与视频解码器及文件格式,并支持 DVD 影音光盘,VCD 影音光盘及各类流式协议。它也能作为 unicast 或 multicast 的流式服务器在 IPv4 或 IPv6 的高速网络连接下使用。它融合了 FFmpeg 计划的解码器与 libdvdcss 程序库使其有播放多媒体文件及加密 DVD 影碟的功能。

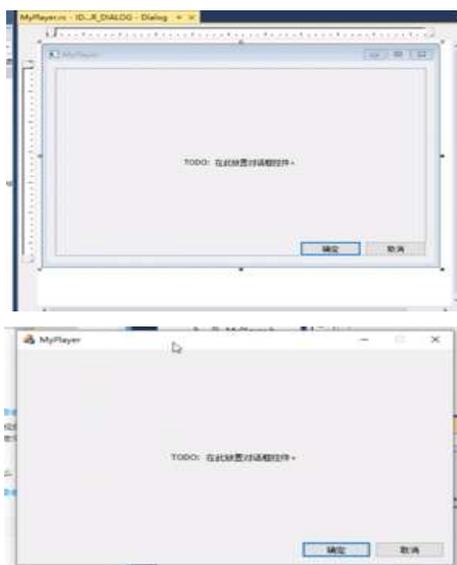
3 设计流程

3.1 MFC 应用

首先利用的是 VS2017 这个软件开发工具,也是我们大学生最常使用的免费的开发工具;通过下载其中的 MFC 插件,进行使用,让我们的代码跑起来过后能够与客户端进行交流。



其中进行到下一步的时候我们选择基于对话框选项，至于对话框大小的话我们就不需要考虑特别的方式，都可以根据自己的喜好进行选择，然后它的都可以不变，不用特殊选择。然后此后就能简单的编译运行了。



会得到这样一个对话框，到这里我们的任务就初步完成了。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。

3.2 功能的选择

大家都知道现在播放器的功能也特别的多，例如：播放、暂停、倍数（快慢）、音量大小、亮度等等。

那笔者这里是由于时间原因呢，没有选择太多的功能，界面的优化也没做到特别的出彩，只能说一个普通的播放器。那笔者做出来的功能呢就只有，播放、暂停、进度条的显示。这也为大家提了个醒，大家能够在时间充裕的情况下多学习是一件特别好也特别难的事情，需要自己把握机会和珍惜机会。

4 功能的实现

4.1 环境的配置

在笔者进行基于 MFC 设计出来的播放器之前，我

们需要配置好我们需要使用的文件、资料以及数据库。所以我们首先需要下载 vs 这个软件开发工具，2007-2017 都可以，但是有细微的差别，有些数据库是后者兼并前者，只要细心的查找就没有太大的差别。

然后是 vs 当中的 MFC 插件，它是一个微软公司提供的类库，我们使用过后不仅大量减少了代码编写量，而且它已经具有一定的功能，我们只需要在他的基础上进行改写和优化了。

当我们创建好一个 MFC 应用后，我们需要在我们的项目里面添加 vlc，它是一个开源的视屏编解码库，我们的视频播放器必须使用到他，其中最关键的就是 vlc.h 这个头文件，里面有我们最核心功能实现的基础。（点开项目属性过后，点击 C++ 属性，添加 VLC 库）

最后就是添加项目连接器中的 lib 静态库，静态库本身就包含了实际执行代码、符号表等等，只需保证在开发者的计算机中有正确的 LIB 文件，在以二进制形式发布程序时不需考虑在用户的计算机上 LIB 文件是否存在及版本问题，可避免 DLL 地狱等问题。

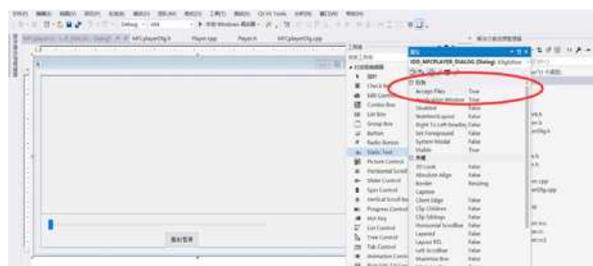
4.2 功能的实现

因为笔者自己的时间原因，所制作的播放器的功能十分的简单，其中就只有播放，暂停，时间条等几个功能。

4.2.1 文件的获取

首先我们应该想到的是我们制作的这个播放器要能够进行视频的播放，那么我们最该先完成的就是视频文件的获取，让我们的视频播放器能够接收到我们的视屏文件，进行辨认后播放。所以我们第一步就是打开我们资源文件中的 .rc 文件，其中一个 Dialog 文件是我们对话框已经运行出来的效果预览图。

选择对话框，右击属性，其中 Accept File 功能我们选择为 True，此后就能看到我们对话框在进行文件拖拽的时候，鼠标下有个+号，这时候我们第一步已经完成了。



4.2.2 播放功能的实现

接下来就是核心播放功能的实现。

首先是在 Dialog 中进行按钮的布局，打开工具箱，选择 Button 按钮，让我们的对话框中有播放按钮。

其次，对该播放按钮的 ID 进行改写，因为我们写程序不仅需要写给自己看，也需要给别人看，比如老师。所以我们 ID 要简单明了表达实现的功能这里笔者打一

个比方 (IDC_BTN_PAUSE)。

```
// 首先初始化
Init();
m_pVlc_media = libvlc_media_new_path(m_pVlc_
Inst,szVedio);// 创建一个新的初始化对象
m_pVlc_player = libvlc_media_player_new_from_
media(m_pVlc_media);// 创建一个播放环境
libvlc_media_release(m_pVlc_media);// 释放 该 媒
体库
libvlc_media_player_play(m_pVlc_player);// 进 行
播放
libvlc_media_player_set_hwnd(m_pVlc_
player,hWnd);// 指定窗口播放句柄
```

这些都是我们 Play 类中的实现功能, 完成后才能进行播放功能。

PS: 需要记住的是我们在进行视频播放时候, 在拖拽的过程中需要带上视屏的路径。// 获取视屏文件路径

```
TCHAR VideoPath[MAX_PATH];
DragQueryFile(hDropInfo, nCount-1, VideoPath,
MAX_PATH);
```

4.2.3 暂停功能的实现

我们的暂停功能与我们的播放功能公用一个按钮。

那么这里也会遇到一个关键的地方, 就是对我们的播放文件的状态进行一个判断, 比如正在播放、没有开始播放等等。这里我们就需要使用定义一个 bool m_bPlayState 表示我们视频的播放状态 (其中 true 表示正在播放, false 表示没有播放)。进行一个简单的判断过后改变他的状态, 从而达到我们暂停功能的实现。

```
if (m_bPlayState == true) // 暂停
{
    m_vedio.Pause();
    m_bPlayState = false;
}
```

最后再调用我们的 libvlc_media_player_set_pause(m_pVlc_player,1); 暂停功能就实现了。

4.2.4 进度条的实现

最后笔者完成的一个功能就是进度条的展示, 以此来提醒观看者我们的视频进行程度。所以我们需要想到的是, 如何获取视频的时间长度, 以及进度条与时间的匹配。

首先我们回到 Dialog 中, 从工具箱中选择 spin Control, 位置随意进行布局。然后我们回到如何获取视频时间长度的步骤, 利用 lib 库中 libvlc_time_t 进行视频时间的捕获。

```
// 获取视频长度
```

```
libvlc_time_t CPlay::GetLength()
{
    libvlc_media_parse(m_pVlc_media); // 解析视屏
资源
    return libvlc_media_get_duration(m_pVlc_media);
// 获取视频长度, 然后返回
}
```

其中我们需要注意的是首先对视屏资源的解析, 解析过后变成我们编译器能够识别的语言后传给我们的编译器, 最后才能实现视屏时间长度的捕获。我们也需要对我们进度条的前进与视屏的播放进行匹配, 所以这个时候我们需要获取当前视屏播放的位置进行绑定, 然后完成进度条和我们视屏播放的时间位置对应, 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。

```
// 获取视频当前长度
libvlc_time_t CPlay::GetCurrentPos()
{
    return libvlc_media_player_get_time(m_pVlc_
player); // 获取视频当前长度,
然后返回
}
```

在获取我们视屏时间和我们进度条进行匹配后我们再完成一个定时器, 那我们整个的进度条功能就完成了

```
void CMFCPlayerDlg::OnTimer(UINT_PTR nIDEvent)//
定时器
```

```
{
    // TODO: 在此添加消息处理程序代码和 / 或调
用默认值

    switch (nIDEvent)
    {
        case 1:
        {
            int nPos = m_vedio.GetCurrentPos();// 获
取当前长度
            m_slider.SetPos(nPos);// 设置控件位置
        }
        default:
            break;
    }

    CDialogEx::OnTimer(nIDEvent);
}
```

4.2.5 界面的布局

这里是整个项目中最简单, 完成一个播放器的布局,

与笔者之前写到的相呼应，首先完成的是一个对话框（包含大小和播放器的名称），其次是我们按钮的布局（播放 / 暂停按钮、进度条），最后是笔者在这里需要补充的是利用 `GetClientRect()` 来获得客户区的大小使用 `FillSolidRect()` 来填充底色。在笔者的代码中体现为：

```
BOOL CMFCplayerDlg::OnEraseBkgnd(CDC* pDC)// 使背景变黑
{
    CRect rect;// 获取矩形区，就是客户端窗口大小
    m_show.GetClientRect(&rect);// 显示区域的大小

    pDC -> FillSolidRect(&rect,RGB(0,0,0));// 填充矩形

    return true;
    //return CDialogEx::OnEraseBkgnd(pDC);// 注释掉，防止再次调取基类
}
```

5 结语

从确定课程设计的题目到现在将近两个月的时间，笔者的“播放器”已初步形成。通过本次设计，又熟悉

了一门新的软件和许多视频解码的知识。在笔者制作这个基于 MFC 播放器的过程中，遇到太多的打击与挫败了，其中最关键的是环境的配置，这个条件是这个播放器制作成功与失败的关键因素，耽误了很多时间，在不断地调试与编译中磨练了自己很多不足的地方；查阅大量资料和事情案例，以及同学和老师的帮助才能完成这个项目。

其实在最开始的时候本人自己也不相信自己能完成这么艰难的挑战，需要自己完成一个项目，这对于以前是前所未有的，所以最初就给自己设置了心里的障碍，越是害怕越是不敢触碰，更不要说能够主动正视这个挑战，随后时间的推移，项目设计已经迫在眉睫，终于接受了这个事实。最后历经“千难万险”完成了本人以前不敢相信的挑战，以此告诫同学们，相信自己，不逼自己一下永远不晓得自己的潜力多大。

【参考文献】

- [1] 赵赛赛. 影视资源在中学语文教学中的应用 [D]. 山东: 科技经济导刊, 2018-05-01.
- [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] 郭泳雪. 软件工程实验教学模式改革与实践研究 [J]. 广东省环境保护职业技术学校, 2020-05-19.