

VC + 6.0 实现语音采集与播放的控件

李 鹏

摘 要 DirectSound 以其强大的音频处理能力, 广泛应用于音频开发。ActiveX 控件是一个软件组件, 可以用在不同的程序中。本文结合 DirectSound 和 ActiveX 控件的优点, 实现了一个 ActiveX 语音控件和网络传输功能。

关键词 DirectX, DirectSound, ActiveX 控件, CasyncSocket

一、引言

DirectX 为 Windows 而设计的 API 允许开发人员调用多媒体硬件设备的各项功能, 通过编程让 Windows 来操作硬件设备。微软推出 DirectX 技术之后, 游戏开发迎来了一场革命, 各个厂商都在说开发的产品支持 DirectX, DirectX 已经成为市场工业的标准。而其中 DirectSound 具有强大的语音处理功能, 给语音的应用带来了极大的方便性。本文先对 DirectSound 的原理进行深入的研究, 并在此基础上, 应用 DirectSound 实现一个语音的 ActiveX 控件。控件实现了全双工的语音通信。

二、DirectX8.1 的概述

DirectX 中大部分 API 都是基于组件对象模型(COM)的。DirectX 中表征设备对象的每个接口都由 IUnknown 接口派生而来, 如 IDirectSound, IDirectInput 等。COM 组件是以 Win32 动态链接库或可执行代码的形式发布。COM 的主要目标是实现类的接口同实现方法分离, 也就是说使用 DirectX 组件不必了解它的具体实现。应用 DirectX 时不可避免会遇到两个与 COM 有关的函数, 一个是用来初始化 COM 环境的函数 CoInitialize(NULL), 另一个是释放 COM 环境的函数 CoUninitialize()^[1]。

三、DirectSound 功能、采集和播放语音的原理

DirectSound 是用来处理声音的 API, 除了播放声音和处理混音之外, 还加强了 3D 音效, 并提供了录音功能。DirectSound 可以加载和播放 wave 文件以及 MIDI 文件, 同时播放复合源 (multiple sources), 合理触发声音事件。可以采集 MIDI 数据或流从一个端口到另一个端口。DirectSound 通过硬件抽象层来访问语音设备, 自动利用语音硬件的加速功能。在当前设备不具备某类特性时, DirectSound 会自动地使用软件来模仿这一功能, DirectSound 是一个完整的 Win32 程序。图 1 是 DirectSound 体系结构图^[3]。

DirectSound 采集和播放语音的简单原理如下: 使用声音

采集对象 DirectSoundCapture 和声音采集缓冲区 DirectSound-CaptureBuffer 来共同完成声音的采集, 使用声音播放缓冲区 DirectSoundBuffer 来存放从网上接收到的语音包, 并用声音播放对象 DirectSound 进行播放^{[1][2]}。

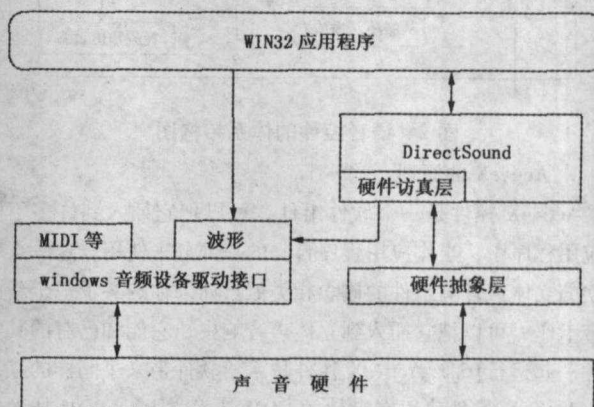


图 1 DirectSoundBuffer 体系结构

DirectSoundBuffer 对象是包含 PCM 格式声音数据的缓冲区, 在播放声音时需要两个缓冲区, 主缓冲区包含用户最终能听见的声音, 从缓冲区存放从 DirectSound 对象采集的声音数据。从缓冲区可存在于整个应用的生命期, 也可以在不用时将其释放, 它可以包含一个重复播放的单独声音, 也可以不时装入新的声音数据来更新声音的输出, 创建一个播放缓冲区, 应该首先构造 DSBUFFERDESC, 这个结构应与采集缓冲区的一样。

播放声音的步骤: 用从播放缓冲区的 lock 方法, 给从缓冲区的一部分加锁, 将声音数据写入主播放缓冲区, 在用从播放缓冲区的 unlock 方法缓冲区解锁, 主播放缓冲区在调用相应的函数进行播放。

一般地, 播放缓冲区在到达末尾时自动停止播放, 但是如果将播放缓冲区对象的 PLAY 方法的 dwFlags 参数设置为 DSBLPLAY_LOOPING, 缓冲区将重复播放直到调用播放缓冲区对象的 Stop 方法^[3]。

四、控件的体系结构、原理及实现

DirectSound 以其对语音处理的方便性而得到了普遍的应用。针对目前在互连网上应用语音已经成为可能,我们设计实现了一个 ActiveX 语音控件。控件用 mfc 的 CasyncSocket 类实现语音包的网路传输。图 2 是语音控件的体系结构图。

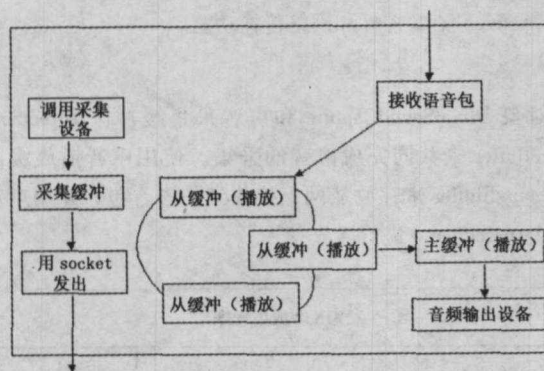


图 2 语音控件的体系结构图

1. ActiveX 的原理

ActiveX 控件是一个软件组件,可以把它插入到许多不同的应用程序中,当作应用程序自身的一部分来使用,这一点和独立的立体声音响组件的概念相类似。如果你购买了一个新的磁带卡座,可以把它插入到立体声音响中,它能和已有的其它部件一起运行[5]。ActiveX 控件技术给软件带来了同样的互用性。ActiveX 控件建立在微软的 COM 技术之上,COM 技术定义了构建 ActiveX 对象的方式及设计 ActiveX 接口的方式。ActiveX 控件还定义了建立于 COM 之上的层面、各种对象应该支持的什么样的接口以及如何与不同类型的对象交互。

ActiveX 控件的关键技术之一是自动。“自动”可以将一个应用程序嵌入到另一个应用程序中,以激活自身并控制其用户接口或文档部分,同时进行自身的更改,然后当用户转移到非所嵌入的应用程序控制的其它部分时,所嵌入的应用程序自动关闭。

完成自动工作的关键 Idispach (也称为调度接口)的特殊接口。Idispach 接口是由指向可用方法列表的指针组成,这些方法可以在 ActiveX 控件中运行或嵌入应用程序中。这些方法都有 ID 号码,称为 DISPID。当知道一个特定方法的 DISPID 之后,就可以通过调用 Idispach 接口的 Invoke 方法来调用该方法,传递 DISPID 以识别所要运行的方法[3][5]。

2. CasyncSocket 的应用

CasyncSocket 对象表示了一个 Windows Socket,即一个网路通讯的终端。CasyncSocket 封装了 Windows Socket API,然而它比直接利用 Windows Socket 编程要简单得多,对于面向对象

的程序员想用 mfc 编程提供了结合点。基类 CasyncSocket 提供了完整的、事件驱动的报路通信,可以创建自己的派生类来捕获和响应网络事件[5]。

应用一个 CasyncSocket 对象之前,必须先调用该对象的 create 方法,创建 TCP 或 UDP 的套接字,创建套接字之后,对于服务器套接字可以开始监听有没有客户连接,连接之后就是发送和接受消息,而客户则开始进行连接[5]。UDP 协议提供高传输效率的数据报服务,适于实时的语音、图像传输,广播消息等网路传输。该控件选择了 UDP 协议实现网路传输。使用 CasyncSocket 类时可以创建 CasyncSocket 类的派生类,创建派生类的主要原因:希望能捕获在特定情况下触发的事件(接收消息及连接完成等情况下触发的事件)。

3. 语音控件的实现

开发者可以用 mfc 来创建 ActiveX 控件,也可以用 atl 创建 ActiveX 控件,两种方法各有利弊,使用 mfc 创建控件,可以用 mfc 的类库,但相比 atl 来说缺乏灵活性。该语音控件是用 mfc 创建的,因为这样就可以使用 mfc 提供的 CasyncSocket 的派生类来处理网络事件,使网路编程变得简单。

用 mfc 的 classwizard 建立一个 ActiveX 工程,在 step by step 中选择所作控件的特性,控件用到了 DirectX8.1sdk 中的语音处理函数,需要在该工程中引入 Directx8.1sdk 中的一些头文件和类库。控件用 CasyncSocket 类派生类的一个对象创建 udp 的套接字,接着创建异步 socket 接收网路传送的语音包,并加入相应的消息映射。

控件实现了两个方法:Record 和 Stop。Record 方法需要输入一个 IP 参数,该参数是网路传输的目的地。控件的大部分代码是用 DirectSound 处理语音。

```

hWnd = this -> m_hWnd;
InitSock();
InitDirectSound();
OnFormatsOK();
CreateOutputBuffer();
StartBuffers(); AfxBeginThread(ComputeThreadProc,
hWnd, THREAD_PRIORITY_NORMAL);
  
```

以上是 Record 中的代码,InitSock()主要是来创建套接字,处理网路事件。InitDirectSound()初始化 DirectSound 对象,OnFormatsOK()初始化采集和播放的格式,CreateOutputBuffer()创建播放缓冲区,StartBuffer()负责采集和播放。AfxBeginThread()创建一个线程,等待 MsgWaitForMultipleObjects(),该事件是采集缓冲区到达一定位置触发的。

DSBPOSITIONNOTIFY 的对象是 Directx8.1 新加入的接口,用来在采集缓冲区设一定的标记位,可以在这些标记位设置一定的事件,当该事件发生时,触发另一些的事件。采集缓冲区是一个循环缓冲,在该缓冲每隔一定位置设定特定事件,

也就是说每当声音样本放入到采集缓冲区到一定的位置时,就触发一定的事件。该事件触发后,先锁定已经有声音样本的缓冲区,将该缓冲区的声音样本用套接子发出,最后将缓冲区向前移动,以准备继续接收,以下是部分代码:

```
g_pDSBCapture->Lock( g_dwNextCaptureOffset, g_dwNotifySize, &pDSCaptureLockedBuffer, &dwDSCaptureLockedBufferSize, NULL, NULL, 0L );
voicepacket voice;
memcpy(voice.clientid, destID, 20);
memcpy(voice.buff, pDSCaptureLockedBuffer, dwDSCaptureLockedBufferSize);
int nResult;
nResult = Sock.SendTo( & voice, dwDSCaptureLockedBufferSize + 20, 1720, destIP); g_pDSBCapture->Unlock( pDSCaptureLockedBuffer, dwDSCaptureLockedBufferSize, NULL, 0 );
g_dwNextCaptureOffset += dwDSCaptureLockedBufferSize;
g_dwNextCaptureOffset %= g_dwCaptureBufferSize;
```

Stop 方法实现的功能:调用该方法释放资源,包括语音资源的释放和 socket 的关闭。

五、结束语

DirectSound 因其强大的语音处理能力而广泛应用于语音

的开发。本文对 DirectSound 采集和播放语音的原理进行了研究,实现了应用 DirectSound 的语音控件。该控件应用在西安交通大学图书馆虚拟咨询台系统中,实现了点对点的语音通信。结合控件的可复用性和 DirectX 的优势,视频控件加入到网络中,使网络的服务更加完善。

参考文献

1. 林丕源,肖德琴,林毅申.局域网声音实时捕捉和传送的实现[J]. 2002, (3): 64-66.
2. 微软公司 DirectX8.1 sdk help[M/CD]. 微软公司 2001, (9).
3. 微软公司 MSDN20001 [M/CD]. 微软公司 2001
4. 李春洪,毛跃奇,陈贵来等.基于 DirectSound 的声音实时仿真研究[J]. 计算机仿真 2002, (5): 47-49
5. 骆长乐. 学用 Visual C++ 6.0[M]. 清华大学出版社. 1999.
6. Philip Taylor. DirectX8.0 简介[EB/OL]. <http://www.tloveyj.com/gamesdesign/direct8/directx%208.0%20简介.htm>

(收稿日期:2003年4月28日)

(上接第 27 页)

```
FileReader fis = new FileReader("邮件列表文件");
BufferedReader dis = new BufferedReader(fis);
int i = 0;
while((fileline = dis.readLine()) != null){
    if((i%10) == 0)
        MimeMessage msg = new MimeMessage(sess);
    .....
    if((i%10) != 0){ //多个邮件地址之间要加","
        maillist = maillist + "," + fileline;
    }else{
        maillist = fileline;
    }
    i++;
    if((i%10) == 0){ //一封邮件在暗抄地址处加 10 个收件人
        msg.addRecipients(Message.RecipientType.BCC, maillist);
        ..... //设置邮件内容
        Transport trans = sess.getTransport(); //创建发送对象
        trans.send(msg); //发送
    }
}
fis.close();
}catch(Exception error){
    System.out.println("Exception: " + error);
}
```

7. 最后的软件界面,包括收件人邮件列表文件,邮件主题,邮件发件人,邮件附件,邮件正文的输入,如下图。

五、结束语

本软件使用 Jcreator Pro 调试通过,在运行时需要安装 jsdk1.4,并且在 classpath 中包含 Javamail.2.jar 和 activation.jar,在 Windows98 下正常运行。如果读者还希望了解更详细的 SMTP 功能或者有更好的邮件群发解决方案请与我联系。

(收稿日期:2003年9月2日)