



DirectX 5/6 高级多媒体程序开发

北京航空航天大学出版社
ISBN 7-81077-172-2

DirectX 5/6
高级多媒体程序
开发

朱春喜 吴佳鲜 编著

北京航空航天大学出版社

DirectX 5/6

高级多媒体程序开发

朱春喜 吴佳鲜 编著

北京航空航天大学出版社

内容简介

DirectX是Windows系统下开发多媒体应用和游戏程序的新技术,用户越来越多,影响也越来越大。本书深入讨论了DirectX 5 技术的各个方面,包括DirectDraw, Direct3D, DirectSound, DirectPlay, DirectInput, DirectSetup和AutoPlay几个部分,并且介绍了Microsoft最新发布的DirectX 6 所包含的新特性,以及如何使用DirectX SDK开发高性能图形/图像应用程序的方法。

本书可供图形、图像、动画、多媒体以及游戏程序开发人员使用,也可作为大专院校师生的参考书。

图书在版编目(CIP)数据

DirectX 5/6 高级多媒体程序开发/朱春喜, 吴佳鲜编.

北京: 北京航空航天大学出版社, 1999.4

ISBN 7 - 81012 - 858 - 2

I. D… II. ①朱… ②吴… III. 图像处理 - 软件, DirectX IV. TP391.4

中国版本图书馆CIP数据核字(1999)第01751号

JS038/11

DirectX 5/6 高级多媒体程序开发

朱春喜 吴佳鲜 编著

责任编辑 王鑑莉 许传安

责任校对 陈 坤

*

北京航空航天大学出版社出版发行

北京市学院路37号(100083) 发行部电话 82317024 发行部传真 82328026

<http://www.buaapress.cn.net>

E-mail: pressell@publica.bj.cninfo.net

北京宏文印刷厂印制 各地书店经销

*

开本: 787×1092 1/16 印张: 17.25 字数: 442 千字

1999年4月第1版 1999年4月第1次印刷 印数: 4000册

ISBN 7 - 81012 - 858 - 2/TP · 327 定价: 27.00 元

前 言

自从计算机诞生以来，人们对计算机游戏就充满了热情。以前对字符型的游戏就乐此不疲，随着时代的发展，人们的要求也越来越高，三维实时游戏和多媒体应用已经成为人们追求的热点。

DOS 系统对程序员的限制较少。在 DOS 操作系统下，任何 C 程序员都可以对计算机的硬件直接操作，甚至在必要的时候使用汇编语言。这样就使得程序能够充分发挥各种硬件的固有特性，提高程序的运行速度。但是，DOS 系统下的程序大部分是依赖于具体硬件的，例如不同的声卡就需要有不同的驱动程序，这就给程序员带来了额外的负担。

Windows 的 GDI 是一个进步。Microsoft 提供了在 Windows 下使用图形的方法。Windows 程序员可以不考虑硬件的具体特性，只需要利用 Windows API 中的图形函数处理当前 DC (Device Content) 即可。但是这些图形函数不能直接访问硬件，都需要通过操作系统，由操作系统来完成绘图工作，尤其不能够将内存中的图形快速拷贝到另一块内存。这对要求速度的实时多媒体应用程序或游戏程序来说显然是不够的。

为了在多媒体、游戏开发市场上处于领先地位，Microsoft 推出了 WinG，也就是 Windows Game 开发软件包。WinG 提供一些直接访问硬件的技术，因此可以提高程序的运行速度。但是 WinG 只能提供 256 种颜色，这对商业软件来说远远不能满足需要。因此，大部分的游戏和多媒体应用仍然局限在 DOS 下，只不过都是使用 32 位的虚拟模式而已。

直到 Windows 95 的推出，游戏和多媒体开发技术才迎来了一个彻底的革命。在吸取了以前的经验和教训之后，Microsoft 推出了游戏和商业多媒体开发技术 DirectX。DirectX 一经问世就成为游戏和多媒体开发市场上的宠儿，得到了众多软件厂商的大力支持，迅速成为市场上的工业标准。人们因此称它为 Windows 下的 Game OS。

DirectX 技术的最大特点是能直接对硬件抽象层 (HAL) 操作，实现视频、声音的输出、网络相互通信，特别能对游戏操纵杆直接编程。与传统的 GDI 和 MCI 相比，不仅大大加快了速度，而且大大地提高了游戏的质量。DirectX 主要包括 DirectDraw, Direct3D, DirectSound, DirectPlay, DirectInput, DirectSetup 六个部分。其中 DirectDraw 管理程序的视频输出；Direct3D 管理程序中三维场景和图形的快速变换；DirectSound 管理程序的声音输出和采集；DirectPlay 管理程序之间的网络通信和信息交换；DirectInput 管理程序的外部输入，包括键盘、鼠标、操纵杆等；DirectSetup 管理程序的安装等。

本书共分九章。第一章是对 DirectX 技术的整体概括，包括 COM 技术介绍；第二章讨论 DirectX 中最重要的 DirectDraw 部分；第三章讲述 Direct3D；第四章讲述 DirectSound；第五章讲述 DirectPlay；第六章讲述 DirectInput；第七章讲述 DirectSetup；第八章讲述 AutoPlay；第九章介绍 DirectX 6 新增加和扩充的特性。

其中第一章至第六章由朱春喜撰写，第七、八、九章由吴佳鲜撰写。

作者的导师许长发教授、李长春教授和王惠珍副教授在作者的写作过程中给予了许多物质和精神上的大力支持，在此对他们表示感谢。作者还感谢众多网友的支持和鼓励。

成书仓促，难免有误，请读者指正。

要想了解 DirectX 技术的最新进展，可以访问下面因特网站点：

<http://www.microsoft.com/directx/default.asp>。

朱春喜 吴佳鲜

于 1998 年 11 月

目 录

第一章 DirectX 入门	1
1.1 DirectX 和部件对象模型 COM	2
1.1.1 部件对象模型 COM (Component Object Model)	2
1.1.2 IUnknown 接口	2
1.1.3 DirectX COM 接口	3
1.1.4 C++和 COM 接口	3
1.1.5 新接口的获取	3
1.1.6 用 C 语言访问 COM 对象	4
1.2 DirectX 5 的基本组成	5
1.2.1 DirectDraw	5
1.2.2 Direct3D	5
1.2.3 DirectSound	5
1.2.4 DirectPlay	5
1.2.5 DirectInput	6
1.2.6 DirectSetup	6
第二章 高性能图形接口——DirectDraw	7
2.1 DirectDraw 的基本图像概念	8
2.1.1 设备无关性位图 DIB (Device-Independent Bitmap)	8
2.1.2 绘图表面 (Drawing Surface)	9
2.1.3 位块传输 Blit	9
2.1.4 页面翻转 (Page Flipping) 和后台缓冲 (Back Buffering)	9
2.1.5 矩 形 (Rectangle)	10
2.1.6 精 灵 (Sprite)	10
2.2 DirectDraw 的体系结构	11
2.2.1 DirectDraw 的对象类型	12
2.2.2 硬件抽象层 HAL (Hardware Abstraction Layer)	12
2.2.3 软件仿真 (Software Emulation)	13
2.2.4 系统组成 (System Integration)	13
2.3 DirectDraw 的组成要素	13
2.3.1 协作等级 (Cooperative Level)	14
2.3.2 显示模式 (Display Modes)	14
2.3.3 DirectDraw 对象	16
2.3.4 表 面 (Surface)	19
2.3.5 调色板 (Palette)	29

2.3.6 剪切板 (Clipper)	30
2.4 DirectDraw 高级特性.....	33
2.4.1 直接内存访问 DMA	33
2.4.2 在窗口模式下使用 DirectDraw 调色板	34
2.4.3 多显示器系统	37
2.4.4 视频端口 (Video Ports)	38
2.4.5 获取翻转和位块传输状态.....	40
2.4.6 检测显示硬件的能力.....	41
2.4.7 在显示内存中存储位图.....	41
2.4.8 三缓冲 (Triple Buffering)	41
2.4.9 DirectDraw 应用程序和窗口风格.....	42
2.5 DirectDraw 例程.....	44
2.5.1 配置 DirectX SDK.....	44
2.5.2 第一个 DirectDraw 实例.....	48
2.5.3 创建动画	60
2.5.4 使用覆盖表面	63
第三章 变幻莫测的三维造——Direct3D	72
3.1 Direct3D 中的基本概念.....	72
3.1.1 动画 (Animation)	72
3.1.2 设备 (Device)	72
3.1.3 颜色模式	73
3.1.4 面	73
3.1.5 帧	73
3.1.6 框架	73
3.1.7 灯光	73
3.1.8 材质	73
3.1.9 阴影	73
3.1.10 纹理	74
3.1.11 用户可视对象.....	74
3.1.12 视点	74
3.1.13 扭曲	74
3.1.14 三维坐标系统.....	75
3.1.15 u 坐标和 v 坐标.....	75
3.1.16 三维变换	75
3.1.17 多边形	75
3.1.18 三角形插值	76
3.1.19 三角形链和扇形.....	77
3.1.20 矢量、节点和四元组.....	77

3.2 Direct3D 保留模式的体系结构	77
3.3 对象和接口	78
3.3.1 对象和索引值	80
3.3.2 Z 缓冲区	80
3.3.3 IDirect3DRM 和 IDirect3DRM2 接口	80
3.3.4 IDirect3DRMAnimation 和 IDirect3DRMAnimationSet 接口	81
3.3.5 IDirect3DRMDevice, IDirect3DRMDevice2 和 IDirect3DRMDeviceArray 接口	81
3.3.6 IDirect3DRMFace 和 IDirect3DRMFaceArray 接口	83
3.3.7 IDirect3DRMFrame, IDirect3DRMFrame2 和 IDirect3DRMFrameArray 接口	83
3.3.8 IDirect3DRMInterpolator 接口	84
3.3.9 IDirect3DRMLight 和 IDirect3DRMLightArray 接口	87
3.3.10 IDirect3DRMMaterial 接口	89
3.3.11 IDirect3DRMMesh, IDirect3DRMMeshBuilder 和 IDirect3DRMMeshBuilder2 接口	89
3.3.12 Direct3DRMObject 对象	90
3.3.13 IDirect3DRMPickedArray 和 IDirect3DRMPicked2Array 接口	90
3.3.14 IDirect3DRMProgressiveMesh 接口	90
3.3.15 IDirect3DRMShadow 接口	91
3.3.16 IDirect3DRMTexture 和 IDirect3DRMTexture2 接口	91
3.3.17 IDirect3DRMUserVisual 接口	92
3.3.18 IDirect3DRMViewport 和 IDirect3DRMViewportArray 接口	92
3.3.19 IDirect3DRMVisual 和 IDirect3DRMVisualArray 接口	92
3.3.20 IDirect3DRMWrap 接口	92
3.4 Direct3D 保留模式例程	93
3.4.1 建立 Hello World	93
3.4.2 预定义和全局变量	94
3.4.3 窗口设置和初始化	96
3.4.4 查找设备驱动程序	102
3.4.5 设置 3D 环境	107
3.4.6 着色循环	109
3.4.7 创建场景	110
第四章 多姿多彩的声音世界——DirectSound	119
4.1 DirectSound 的体系结构	119
4.1.1 回放 (Playback)	120
4.1.2 捕获 (Capture)	120
4.1.3 属性集 (Property Sets)	120
4.1.4 硬件抽象层和硬件仿真层	121
4.1.5 系统组成	121

4.2 DirectSound 的基本要素	122
4.2.1 DirectSound 设备	122
4.2.2 DirectSound 缓冲区	126
4.2.3 3D 声音	136
4.2.4 DirectSound 3D 缓冲区	139
4.2.5 DirectSound 3D 听众	141
4.2.6 DirectSound 声音捕获	142
4.2.7 优化声音特性	147
第五章 合作与对抗——DirectPlay	150
5.1 编写网络程序	150
5.2 DirectPlay 概述	151
5.2.1 DirectPlay 的体系结构	151
5.2.2 会话管理	152
5.2.3 Player 管理	153
5.2.4 消息管理	153
5.2.5 组管理	154
5.2.6 DirectPlay 通信	155
5.3 DirectPlay Lobby	158
5.3.1 DirectPlay Lobby 的结构	159
5.3.2 Lobby 会话	159
5.3.3 Lobby 漫游	161
5.3.4 同步启动	161
5.4 DirectPlay 提供者	161
5.4.1 服务提供者	162
5.4.2 Lobby 提供者	163
5.5 使用 DirectPlay	163
5.5.1 DirectPlay 接口	163
5.5.2 使用回调函数	164
5.5.3 建立 Lobby 应用程序	165
5.5.4 DirectPlay 消息	167
5.5.5 DirectPlay 地址	169
5.5.6 DirectPlay 工具与范例	171
5.5.7 安全与认证	173
5.6 DirectPlay 接口	175
5.6.1 IDirectPlay 接口	175
5.6.2 IDirectPlay2 接口	176
5.6.3 IDirectPlay3 接口	176
5.6.4 IDirectPlayLobby 和 IDirectPlayLobby2 接口	176

5.7 DirectPlay 例程	176
5.7.1 使用 Lobby 建立连接	177
5.7.2 使用对话框连接	179
5.7.3 屏蔽服务提供者对话框	184
5.7.4 创建自刷新会话列表	188
第六章 灵活的交互方式——DirectInput	192
6.1 DirectInput 的结构	192
6.1.1 DirectInput 对象	192
6.1.2 DirectInputDevice 对象	193
6.1.3 DirectInput 设备对象实例	193
6.1.4 DirectInputEffect 对象	193
6.1.5 与操作系统的集成	193
6.2 DirectInput 的组成	194
6.2.1 DirectInput 设备查询	194
6.2.2 DirectInput 设备	195
6.2.3 DirectInput 设备数据	202
6.2.4 力反馈设备	210
6.3 DirectInput 键盘程序设计	225
6.3.1 创建 DirectInput 对象	225
6.3.2 创建 DirectInput 键盘设备	226
6.3.3 设置键盘数据格式	226
6.3.4 设置键盘程序的协作等级	227
6.3.5 访问键盘	227
6.3.6 从键盘获取数据	227
6.3.7 关闭 DirectInput 系统	228
6.4 DirectInput 鼠标程序设计	231
6.4.1 创建 DirectInput 鼠标设备	231
6.4.2 设置鼠标数据格式	231
6.4.3 设置鼠标的行为	232
6.4.4 设置鼠标输入缓冲区	232
6.4.5 管理对鼠标的访问	234
6.4.6 获取缓冲数据	234
6.5 DirectInput 操纵杆程序设计	237
6.5.1 查询操纵杆设备	237
6.5.2 创建操纵杆设备	237
6.5.3 设置操纵杆数据格式	238
6.5.4 设置操纵杆的行为	239
6.5.5 访问操纵杆	240

6.5.6 获取操纵杆输入数据.....	241
6.6 DirectInput 力反馈设备程序设计	243
6.6.1 查询力反馈设备.....	243
6.6.2 创建力反馈设备.....	244
6.6.3 查询设备支持的效果.....	246
6.6.4 创建效果	247
6.6.5 执行效果	249
6.6.6 改变效果	249
第七章 简便易行的安装——DirectSetup	251
7.1 DirectXSetup 的安装过程.....	251
7.2 使用回调函数定制安装.....	252
7.3 安装程序的设计	255
7.4 测试安装程序	257
第八章 自动运行的捷径——AutoPlay	258
第九章 DirectX 6 的新特性	262
9.1 DirectDraw 6.0.....	262
9.1.1 IDirectDraw4 接口	262
9.1.2 获取 IDirectDrawSurface4 接口	263
9.1.3 创建 DirectDraw 对象.....	263
9.2 Direct3D 6.....	264
9.3 DirectPlay 6	265

第一章

DirectX 入门

Microsoft 公司的 DirectX 的推出是一个极好的消息。它包含了编制高级计算机游戏和多媒体应用程序的最新技术和工具，为广大的程序员提供了一整套的应用程序接口 API，使程序员能够设计出高性能实时的应用程序。DirectX 使运行在 Microsoft Windows 操作系统下的游戏比运行在 MS-DOS 操作系统下的具有更高的性能和更好的交互性。

Microsoft 开发 DirectX 的首要目的就是要让图形/图像、动画、多媒体以及游戏程序开发人员在 Windows 环境下很容易地编制高级程序。在此之前，个人电脑上的大部分游戏都是基于 MS-DOS 的，游戏开发人员必须针对不同的硬件接口卡编制不同的程序。尽管代码类似，仍不可避免许多重复劳动。利用 DirectX，可以获得 DirectX 的硬件独立性的优点，同时又可以直接访问硬件。DirectX 可以提供 MS-DOS 中直接的硬件访问特性，同时又去掉了个人电脑中添加新硬件时所带来的硬件识别问题。

因为硬件的升级是不可避免的，所以 DirectX 实际上也提供了一种即插即用的方法。因此，利用 DirectX 可以开发出高性能实时的应用程序，能直接访问计算机中的硬件和将来系统中可能会具有的硬件设备。DirectX 在硬件和应用之间提供了一致的接口以减少安装和配置的复杂性，并且使得对硬件的利用达到最优。利用 DirectX 提供的接口，程序员能充分利用硬件的特性而不需要考虑其具体细节。一个优秀的、基于 Windows 的高性能为多媒体软件等应该能够利用系统的如下技术：

- 为提高性能而专门设计的加速卡；
- 即插即用 PnP (Plug and Play)；
- 内嵌于 Windows 的通信服务，包括 DirectPlay。

DirectX 主要包含有六个部分。下面对各部分作简要说明。

□ **DirectDraw** 通过支持访问屏外显示内存中位图的软硬件加速技术，可利用硬件的位块传输和缓冲区翻转功能快速直接存取。

□ **DirectSound** 提供软硬件声音混合和录音再生功能。

□ **DirectPlay** 使游戏在调制解调器和网络之间的连接更加简单方便。

□ **Direct3D** 提供了高级保留模式 (Retained-Mode) 接口和低级即时模式 (Immediate-Mode) 接口。前者允许程序很容易地完成一个完全的三维图形系统，后者使程序能完全控制着色管道。

□ **DirectInput** 提供了基于 Windows 图形/图像、游戏输入的 API 和驱动程序，它不仅支持目前的键盘、鼠标和操纵杆，也支持将来的基于 Windows 的输入设备。

□ **DirectSetup** 提供了 DirectX 的一次性安装过程。

AutoPlay 是 Windows 95 的特性。当你将光盘放入光驱时，它能自动运行安装程序或游戏。事实上，AutoPlay 是 Microsoft Win32 API 的一部分，只不过为了方便起见也被收到了

DirectX SDK 中。

通过查询各种 DirectX 对象接口，可以检测系统中安装的是 DirectX 的哪个版本。DirectX SDK 的范例文件 GetDXVersion.cpp 中的 GetDXVersion 函数可以完成这一任务。该函数创建了一些关键的 DirectX 对象——一个 DirectDraw 对象、一个 DirectDrawSurface 对象和一个 DirectInput 对象——来检测安装在系统中的 DirectX 的版本。如果系统中安装了老版本的 DirectX SDK，必须确保不要包含它的 Include 和 Lib 路径。使用 DirectX 5 头文件编译后的代码同 DirectX 3 的库链接后将会出现不可预知的错误。

因为 C++ 语言更能对计算机硬件作直接访问，所以 DirectX 采用了 C++ 语言作为开发的主语言。而 DirectX 又是专门为 Visual C++ 设计的，所以编写 DirectX 程序最好能在 Visual C++ 下进行。当然 Borland C++4.5 以上的版本也没有问题，只是配置有些不同罢了，以后的章节将会提到这一问题。Microsoft 还考虑到大量的 Watcom C++ 用户，DirectX SDK 中的几乎每个例程都提供了 Watcom C++ 的 MAKE 文件。为了降低编程的难度，一些公司也针对 DirectX 开发了运行在不同编程环境中的控件，如 Arakelian Soft 公司的 ActiveX 控件 DirectStudio 98 for Visual Basic 5.0、Tegosoftware 公司的 TegoSoft ActiveX for Visual Basic，还有 John Pullen，Paul Bearne 和 Jeff Kurtz 开发的运行在 Delphi 下的 Delphi Games Creator 等。

1.1 DirectX 和部件对象模型 COM

1.1.1 部件对象模型 COM (Component Object Model)

DirectX 中的大部分 API 都由基于 COM 的对象和接口组成。COM 是接口重利用的基于对象的系统的基础，是 COM 编程的核心模型，它也是一种接口规范。在操作系统级别上，它又是一个对象模型。

许多 DirectX API 都有创建 COM 对象的实例。可以将一个对象看做一个黑盒子，对象通过接口与对象通信。通过 COM 接口发送给对象或从对象接收的命令称为方法。例如，IDirectDraw2::GetDisplayMode 方法是通过 IDirectDraw2 接口从 DirectDraw 对象获得当前的显示模式。对象在运行时可以同其它对象捆绑在一起，并且能够使用这些对象的接口。如果已知某个对象是 COM 对象，并且知道该对象支持的接口，则应用程序或其它对象就能够确定第一个对象所能执行的服务。所有 COM 对象都继承一个称为 Query 的接口方法，它可以使你查询一个对象支持的接口以及创建这些接口的指针。

1.1.2 IUnknown 接口

所有的 COM 接口都由一个称为 IUnknown 的接口派生而来。该接口为 DirectX 提供了对象生存期的控制和操作多接口的能力。IUnknown 含有三个方法：

- **AddRef** 当一个接口或另一个应用捆绑到某一个对象上时，就使用 AddRef 方法将该对象的索引值加 1。
- **QueryInterface** 通过指向特定接口的指针查询对象所支持的特性。
- **Release** 将对象的索引值减 1。当索引值变为 0 时，该对象就从内存中释放。

其中 `AddRef` 和 `Release` 方法负责维护对象的索引值。例如，如果创建一个 `DirectDrawSurface` 对象，该对象的索引值就被设定为 1。每当有一个函数返回该对象的指针时，该函数必须通过返回指针来调用 `AddRef` 方法，将该对象的索引值加 1。每一个 `AddRef` 的调用都必须有一个 `Release` 的调用与其对应。当对象的索引值达到 0 时，该对象就被撤销，该对象的所有接口都不可再用。

`QueryInterface` 方法测定一个对象是否支持指定的接口。如果支持，`QueryInterface` 就返回指向该接口的指针。然后可以使用该接口包含的方法同对象通信。如果 `QueryInterface` 成功地返回接口的指针，它就会自动调用 `AddRef` 方法增加对象的索引值。在撤销接口指针之前必须调用 `Release` 来减少对象的索引值。

1.1.3 DirectX COM 接口

DirectX 中的接口是用最基本的 COM 编程创建的。表征设备对象的每个接口都由 `IUnknown` COM 接口派生而来，如 `IDirectDraw2`，`IDirectSound` 和 `IDirectPlay` 都是这样。基本对象的创建工作由动态链接库 DLL 中的特殊函数来处理。

比较一般的情况是，DirectX 对象模型为每个设备提供了一个主对象，其它支持服务的对象由主对象派生而来。例如，`DirectDraw` 对象就表征了显示适配器。你可以利用 `DirectDraw` 来创建表征显示内存的 `DirectDrawSurface` 对象和表征硬件调色板的 `DirectDrawPalette` 对象。同理，`DirectSound` 对象代表了声卡，利用它可创建代表音源的 `DirectSoundBuffer` 对象。

除了能够产生子对象外，设备的主对象还能测定它所表征的设备特性，如屏幕的大小和颜色数、声卡是否支持波表合成等。

1.1.4 C++和 COM 接口

对 C++ 程序员来说，COM 接口就像一个抽象的基类。在 C++ 的基类中，所有的方法都定义为纯虚的，这就意味着没有任何代码同方法关联在一起。纯虚的 C++ 函数和 COM 接口都使用一种称为虚表（vtable）的设备。一个虚表包含了应用于所给接口的所有函数的声明。如果想让程序或对象使用这些函数，可以先用 `QueryInterface` 方法检查对象存在的接口，获取该接口的指针。调用 `QueryInterface` 后，应用或对象实际上就从对象中接收到了虚表的指针。通过该指针就可以调用应用于该对象的所有接口方法。

COM 对象和 C++ 的另一个相似之处是方法的第一个参数就是接口或类的名字，在 C++ 中称为 `this` 参数。因为 COM 对象和 C++ 对象是完全二进兼容的，编译器就将 COM 接口同 C++ 抽象类同样处理，并且具有相同的语法，这会使得代码简单一些。

1.1.5 新接口的获取

部件对象模型不是通过改变存在于接口的方法，而是通过扩展包含有新特性的新接口来更新对象的功能。在保留已有接口状态的情况下，COM 对象能够自由扩展服务并维持同旧的应用兼容。DirectX 部件遵循这一原则。例如 `DirectDraw` 部件支持 `IDirectDrawSurface` 接口的三个版本：`IDirectDrawSurface`，`IDirectDrawSurface2` 和 `IDirectDrawSurface3`。要想利用新接口提供的特性，必须调用对象的 `IUnknown::QueryInterface` 方法，指定欲获取的接口的

全局统一标识符 GUID (globally unique identifier)。接口的 GUID 在相关的头文件中声明。下面的例子说明了如何查询一个新的接口：

```
LPDIRECTDRAW    lpDD1;
LPDIRECTDRAW2   lpDD2;

ddrval = DirectDrawCreate( NULL, &lpDD1, NULL );
if( FAILED(ddrval))
    goto ERROROUT;

// Query for the IDirectDraw2  Interface
ddrval = lpDD1->Query 接口(IID_IDirectDraw2, (void **)&lpDD2);
if( FAILED(ddrval))
    goto ERROROUT;

// Now that we have an IDirectDraw2, release the original Interface
lpDD1->Release();
```

一般来说，新的接口都会支持以前版本的接口。不过也有例外，IDirect3DDevice2 接口就是特例。如果应用程序需要早期版本的接口所提供的特性，可以利用老接口的 GUID，使用上述例程来查询该接口。

1.1.6 用 C 语言访问 COM 对象

使用 C 语言也可以调用任何 COM 接口方法。用 C 语言调用一个接口方法需要注意两件事：

- 方法的第一个参数始终是已经创建的调用该方法的对象（即 this）；
- 接口中的每个方法都通过对象虚表的指针来引用。

下面的例子使用 C 语言调用 IDirectDraw2::CreateSurface 方法创建一个同 DirectDraw 对象关联的表面：

```
ret = lpDD->lpVtbl->CreateSurface (lpDD, &ddsd, &lpDDS,
    NULL);
```

其中 lpDD 参数是同新表面关联的 DirectDraw 对象，同时该方法对一个描述表面（Surface）的结构&ddsd 赋值，并返回指向新接口&lpDDS 的指针。调用 IDirectDraw2::CreateSurface 方法，首先要指向该 DirectDraw 对象的虚表，然后获取虚表中的方法。方法中提供的第一个参数是已经创建的 DirectDraw 对象的索引。下面的一行代码说明了 C 和 C++对 COM 对象方法调用的不同之处：

```
ret = lpDD->CreateSurface(&ddsd, &lpDDS, NULL)
```

1.2 DirectX 5 的基本组成

大家知道, Microsoft 一直在对 DirectX 升级, 从 DirectX, DirectX2, DirectX3 到 DirectX 5, DirectX 6, 马上又会推出 DirectX 7。事实上, DirectX 5 已经包含了 DirectX 几乎所有的功能。本书以 DirectX 5 为主来介绍。相对于 DirectX 3 来说, DirectX 5 提供了不少新的功能和服务, 不过利用 DirectX3 SDK 写的程序不需要修改就可以成功地编译和运行。同样, 用 Direct5 SDK 写的程序也可平滑地移植到 Direct6 SDK 上。

1.2.1 DirectDraw

DirectDraw 扩展了新的视频端口能力, 允许应用程序控制从硬件视频端口传送到显示内存中的 DirectDraw 表面的数据流。另外, DirectDraw 硬件仿真层 HEL (Hardware Emulate Layer) 能够利用 Pentium MMX 处理器提供的加速性能。DirectDraw 在第一次创建表面时会检测处理器是否是 MMX 处理器。在非奔腾处理器的计算机上, 该测试会导致调试器报告出一个事件, 当然该事件不会影响程序的性能和稳定性。DirectDraw 支持比主表面 (Primary Surface) 更宽的屏外表面 (Off-screen Surface)。只要硬件允许, 你可以创建足够宽的表面。

DirectDraw 支持加速图形端口 AGP (Accelerate Graphics Port) 特性。在一个 AGP 设备系统, 你可以在非本地显示内存创建表面。DDSCAPS 结构支持标准 (本地) 显示内存和 AGP (非本地) 显示内存之间不同的标志。DirectDraw 可以向非本地显示内存位块传输 DDSCAPS 结构包含的成员信息。

1.2.2 Direct3D

Direct3D 即时模式 (Immediate Mode) 现在支持 Drawing Primitive, 而不必直接在缓冲区中工作。Direct3D 保留模式 (Retained Mode) 支持插值, 使你能够调和颜色、平滑地移动物体、变形网格, 并且执行许多其它的变换。保留模式也支持渐进网格, 即允许你从一个粗糙网格开始, 然后再细化它。

1.2.3 DirectSound

DirectSound 包含了一个新的接口 IksPropertySet, 能够支持由声卡和相关驱动程序提供的扩展服务。基于 COM 的函数 DirectSoundCapture 也是新的, 它可以直接访问驱动程序。

1.2.4 DirectPlay

DirectPlay 包含了一个新的接口 IDirectPlay3, 除了新的方法之外, 其它同 IDirectPlay2 完全相同。同样, IDirectPlayLobby2 也是 IDirectPlayLobby 的升级版本。DirectPlay 的新功能包括通过创建连接快捷方式来隐藏服务提供者 (Service Provider) 对话的能力; 还可以保持可用会话的更新列表, 应用 SetSessionDesc 方法更好地支持口令保护会话, 支持保密服务连接。使用 CoCreateInstance 还可直接创建多个 DirectPlay 对象。

1.2.5 DirectInput

DirectInput 提供了操纵杆的 COM 接口，同时还提供了键盘和鼠标以及力反馈装置的 COM 接口。

1.2.6 DirectSetup

DirectSetup 包含了更多的用户接口定制能力，由一个回调函数提供。该回调函数将当前的安装状态传递给安装程序。可以通过定制的用户接口利用这一信息来显示安装状态。另外，DirectSetup 提供了多个游戏玩家的操作方法。在软件中，开发人员可以使用 DirectPlayLobby 从注册列表中去掉注册信息。