HZ BOOKS
华章教育

DIGITAL 数字媒体专业规划教材

Beginning Game Programming with DirectX

DirectX 游戏编程 基础教程

王德才 等编著

机械工业出版社
China Machine Press

DIGITAL 数字媒体专业规划教材

Beginning Game Programming with DirectX

DirectX 游戏编程基础教程

王德才 等编著

机械工业出版社
China Machine Press

本书系统全面地介绍了DirectX编程的各个方面，主要内容包括：学习DirectX之前必须掌握的基础知识、DirectX与相关图形技术以及Windows系统本身的关系、Direct3D三维图形和动画开发、DirectX Audio音频开发、DirectInput输入处理以及DXUT程序框架等内容。此外，每章都提供了精心设计的示例程序和课后练习，以及相应的源代码和多媒体课件。

本书非常适合作为高等院校相关专业的教材。对于希望进入DirectX游戏开发领域的人员，本书是一本非常好的自学教材。对于希望了解新一代Windows操作系统（Windows Vista和Windows7）图形引擎，以及新一代图形界面开发技术（WPF）底层基础的开发人员，本书也是一本优秀的参考用书。

封底无防伪标均为盗版
版权所有，侵权必究
本书法律顾问　北京市展达律师事务所

图书在版编目（CIP）数据

DirectX游戏编程基础教程 / 王德才等编著. —北京：机械工业出版社，2010.9
（数字媒体专业规划教材）

ISBN 978-7-111-31561-2

Ⅰ. D…　Ⅱ. 王…　Ⅲ. ① 多媒体—软件工程，DirectX—教材　② 游戏—应用程序—程序设计—教材　Ⅳ. ① TP311.56　② G899

中国版本图书馆CIP数据核字（2010）第157473号

机械工业出版社（北京市西城区百万庄大街22号　邮政编码　100037）
责任编辑：陈佳媛
三河市明辉印装有限公司印刷
2010年9月第1版第1次印刷
185mm × 260mm · 18.5印张
标准书号：ISBN 978-7-111-31561-2
定价：35.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88378991；88361066
购书热线：（010）68326294；88379649；68995259
投稿热线：（010）88379604
读者信箱：hzjsj@hzbook.com

前　言

DirectX最初是微软为了弥补Windows 3.1对图形、声音处理能力不足的缺陷，开发的一个附加工具包，主要用于在Windows平台上开发游戏。自从DirectX 1.0发布以来，DirectX的版本不断升级、功能逐渐完善，很快就成为对多媒体系统各个方面都有决定性影响的开发接口。目前DirectX主要由负责三维图形和动画开发的DirectX Graphics、负责音频开发的DirectX Audio以及支持各种输入设备的DirectX Input三部分组成，这三部分各自又包含众多功能组件。

在DirectX的所有功能组件中，负责3D图形和动画开发的Direct3D是最重要、最复杂、也是更新最快的组件。随着Direct3D版本的不断升级，其功能也越来越强大，图形硬件对Direct3D的支持也越来越好。往往在新版本的Direct3D刚刚发布后，硬件厂商就会生产出支持最新版本Direct3D功能的显卡。

现在DirectX已经不仅仅是一个用于游戏以及高性能多媒体应用程序开发的附加工具包，而是整体Windows系统的图形基础，成为Windows系统自身的重要组成部分。在最新的Windows Vista以及Windows 7操作系统中，使用Direct3D作为底层的图形引擎，而放弃了自从Windows 3.1以来一直使用的GDI/GDI+。甚至最新的图形界面开发技术WPF在底层也是基于Direct3D开发的，Direct3D也逐渐成为用户界面开发技术的底层基础。因此DirectX编程已经不仅仅是游戏以及高性能多媒体应用开发人员所必须具备的技能，而成为所有Windows开发人员需要了解的技术基础。

本书主要面向首次接触DirectX的初级开发人员和在校师生。由于DirectX编程相对比较复杂，为了使本书能够成为一本适合DirectX开发入门的优秀图书，在本书的写作过程中，作者有意突出了以下特点。

1. 内容系统全面、逻辑性强。由于DirectX本身比较复杂，而且需要各种相关基础知识，为了使读者的学习曲线尽可能平滑，一方面精心选择学习内容，另一方面确保各章内容循序渐进，使读者能够顺利进入DirectX开发领域。对于那些较高级的重点内容，以选读的形式给出，以方便读者依据个人情况选择学习内容。为了便于教学，还为教师免费提供配套的多媒体课件，教师可从华章网站（www.hzbook.com）下载。

2. 精心设计大量的示例程序和课后练习。针对具体的开发技术，作者精心设计了相应的示例程序，并在源代码中给出了详细注释，一方面可以帮助读者加深理解，另一方面也便于读者进行扩展，逐步提高自身的开发水平。所有示例程序均可以从华章网站下载。此外，为了引导读者深入学习并加深对所学内容的理解，每章还提供了相应的课后练习。

3. 提供大量的提示和说明。对于一些比较难理解和容易犯错的地方，作者结合多年的教学和开发经验，提供了相应的提示和说明。所以对于已经掌握了DirectX开发的人员，本书也具有一定的参考价值。

本书内容涵盖DirectX开发的各个方面，重点在介绍DirectX开发技术本身，对于应用开发有所涉及但是还不够深入，如果将来准备专门从事大型游戏以及复杂可视化仿真系统的开发，还需要进一步阅读相关资料，不过阅读本书可以为后续深入学习奠定坚实的基础。对于准备使用DirectX进行简单的应用开发和希望了解Windows图形技术基础的开发人员，本书提供的内容就已经足够了。

本书主要由王德才撰写。吴明飞、唐业军、胡强、余华鸿、孙牧、刘渊、戴君、刘玉达、姜晓佳、周晓敏、姜少孟等也参加了部分章节的撰写以及资料整理、图形绘制、校订等工作。最后还要感谢为本书出版付出辛勤劳动的华章编辑们。

在本书编写过程中我们力求精益求精，力图为读者奉献一本优秀的书籍，使本书成为学习DirectX开发的优秀教材。但是由于作者水平有限，书中难免有不足之处，敬请读者批评指正。

作　者

2010年7月

此为试读，需要完整PDF请访问：www.ertongbook.com

教学建议

教学内容	学习要点及教学要求	课时安排	
		讲课学时	实验学时
第1章 DirectX与游戏编程简介	• 了解DirectX的基本概念与技术背景 • 了解DirectX与OpenGL、XAN以及WPF之间的关系 • 掌握DirectX的组成 • 掌握DirectX的安装与配置 • 理解DirectX与游戏开发之间的关系	2	
第2章 预备知识	• 掌握向量、矩阵以及平面的概念与基本运算 • 熟悉D3DX提供的数学运算函数 • 掌握Win32 API程序的开发步骤 • 理解Win32 API程序的运行机理 • 掌握COM对象的使用	2~4	
第3章 Direct3D编程入门	• 了解Direct3D的体系结构 • 掌握Direct3D表面的概念及相关内容 • 掌握Direct3D设备的有关内容及其初始化过程 • 了解Direct3D 9Ex扩展编程	2	1
第4章 基本图元和文本绘制	• 理解Direct3D坐标系 • 了解Direct3D基本图元类型 • 理解Direct3D中的颜色表示 • 掌握如何使用顶点缓存和索引缓存绘制图形 • 掌握Direct3D中二维文本的绘制 • 理解Direct3D渲染状态 • 了解图形反锯齿	4	1
第5章 顶点坐标变换	• 理解Direct3D顶点变换与光照流水线 • 理解Direct3D矩阵类型 • 掌握Direct3D顶点变换的具体实现	2	1
第6章 光照	• 理解Direct3D光照计算模型 • 理解Direct3D光源与材质 • 掌握Direct3D基本光照效果的实现	2	1
第7章 纹理映射	• 理解纹理的相关概念 • 掌握纹理对象的创建 • 理解纹理过滤方式和寻址模式 • 理解纹理阶段混合 • 了解多层纹理映射	2~3	1
第8章 Direct3D渲染技巧	• 理解Alpha混合的基本原理 • 掌握使用Alpha混合实现半透明效果 • 理解Direct3D雾化原理 • 掌握Direct3D雾化效果的实现 • 了解Direct3D模板测试的基本过程与基本原理 • 了解Direct3D模板测试的应用	2~3	1

（续）

教学内容	学习要点及教学要求	课时安排	
		讲课学时	实验学时
第9章 网格模型	• 理解Direct3D网格模型基础 • 了解简单几何体和三维文本网格模型 • 掌握文件网格模型的加载和渲染 • 理解网格模型优化 • 掌握网格模型类 • 理解物体空间状态的控制 • 了解蒙皮骨骼动画网格模型	4~6	2
第10章 点精灵与粒子系统	• 理解粒子系统的基本原理 • 理解粒子与点精灵 • 掌握粒子系统的典型实现 • 了解粒子系统的不同实现方式	2	2
第11章 基本虚拟场景构造	• 理解三维地形模拟的基本原理 • 掌握基本高度图地形模拟 • 掌握简化的天空模拟方法 • 理解虚拟场景漫游的基本原理 • 理解虚拟摄像机类及其实现	4	2
第12章 HLSL高级着色语言	• 掌握HLSL数据类型 • 掌握HLSL运算符和语句 • 理解HLSL函数 • 了解HLSL语义 • 掌握编写HLSL着色器的基本步骤	2	
第13章 着色器和效果	• 理解顶点着色器的基本概念和顶点声明 • 掌握使用顶点着色器的基本步骤 • 理解使用顶点着色器实现基本光照计算模型 • 理解像素着色器的基本概念 • 掌握使用像素着色器的基本步骤 • 理解使用像素着色器实现纯亮度显示 • 理解效果的基本概念与使用 • 了解使用效果管理状态	4~6	2
第14章 DirectX Audio音频编程	• 了解DirectX Audio的技术架构及其支持的音频格式 • 理解XACT的组成以及基本概念 • 了解使用XACT GUI制作声音的基本步骤 • 掌握使用XACT播放声音的基本步骤 • 了解零潜伏时间流 • 了解代码驱动的XACT API • 理解XAudio2的基本概念 • 掌握XAudio2的初始化以及使用XAudio2播放声音的基本步骤 • 了解资源交换文件格式以及如何使用XAudio2加载音频数据	4	2
第15章 DirectInput与场景交互	• 理解DirectInput的初始化 • 掌握基本的键盘、鼠标输入处理 • 理解游戏杆输入处理 • 了解拾取的基本原理和具体实现	2~3	1
第16章 DXUT程序框架	• 理解DXUT框架的组成、功能以及基本原理 • 掌握使用DXUT框架的基本步骤 • 掌握如何使用DXUT框架渲染文本和控件 • 了解如何基于DXUT框架进行应用开发	2~3	1
	教学总学时建议	42~52	18

说明 1 本教材可作为数字媒体、游戏开发、计算机等专业的游戏开发或多媒体开发等课程的教材，理论授课学时数为42~52学时（相关配套实验单独安排，共18学时），不同专业根据不同的教学要求和计划学时数可酌情对教材内容进行适当取舍，其中可以删减的内容作为选读章节用星号(*)在书中进行了标识。

2 本教材理论授课学时包含习题课、课堂讨论等必要的课内教学环节。

目　录

前言

教学建议

第1章　DirectX与游戏编程简介 ……1

1.1　DirectX介绍 ……1

*1.2　OpenGL、XNA与WPF ……2

1.2.1　OpenGL ……2

1.2.2　XNA ……2

1.2.3　WPF ……3

1.3　DirectX功能组件 ……4

1.3.1　DirectX Graphics ……5

1.3.2　DirectX Audio ……5

1.3.3　DirectX Input ……6

1.3.4　其他组件 ……6

1.4　DirectX与游戏开发 ……7

1.5　DirectX安装与配置 ……7

1.5.1　系统配置要求 ……7

1.5.2　DirectX安装 ……8

1.5.3　选择调试库和发布库 ……8

1.5.4　在Visual Studio IDE中配置DirectX ……9

1.5.5　浏览DirectX SDK示例程序 ……10

1.6　小结 ……11

练习 ……12

第2章　预备知识 ……13

2.1　数学基础 ……13

2.1.1　向量 ……13

2.1.2　矩阵 ……15

2.1.3　平面 ……18

2.2　Win32 API编程基础 ……19

2.2.1　开发Win32 API程序的基本步骤 ……20

2.2.2　Win32 API程序解析 ……23

2.3　COM使用基础 ……28

2.3.1　COM对象概述 ……28

2.3.2　创建COM对象 ……30

2.3.3　使用COM接口 ……30

2.3.4　管理COM对象的生命期 ……31

2.4　小结 ……31

练习 ……32

第3章　Direct3D编程入门 ……33

3.1　Direct3D体系结构 ……33

3.1.1　硬件抽象层 ……33

3.1.2　软件参考层 ……33

3.1.3　Direct3D系统集成 ……34

3.2　Direct3D表面 ……34

3.2.1　IDirect3DSurface接口 ……35

3.2.2　宽度和间距 ……35

3.2.3　像素格式 ……35

3.2.4　交换链与表面翻转 ……35

3.3　Direct3D设备对象 ……36

3.3.1　Direct3D设备类型 ……36

3.3.2　顶点运算 ……37

3.3.3　创建Direct3D设备对象 ……37

3.3.4　设备丢失与恢复 ……41

3.4　最简单的Direct3D程序 ……41

3.4.1　程序基本结构 ……42

3.4.2　常用宏定义 ……42

3.4.3　初始化Direct3D ……42

3.4.4　渲染图形 ……43

3.4.5　结束Direct3D程序 ……45

*3.5　Direct3D 9Ex扩展编程 ……45

3.5.1　WDDM和Direct3D 9Ex介绍 ……45

注：加星号的章节为选读内容。

3.5.2 Direct3D 9Ex初始化 ……46
3.6 小结 ……47
练习……47
第4章 基本图元和文本绘制 ……48
4.1 Direct3D坐标系……48
4.2 Direct3D基本图元……48
4.3 Direct3D中的颜色表示……49
4.3.1 使用D3DCOLOR定义颜色 ……50
4.3.2 使用D3DCOLORVALUE定义颜色 ……50
4.3.3 使用D3DXCOLOR定义颜色 …50
4.4 使用顶点缓存绘制图形 ……51
4.4.1 顶点结构和顶点格式 ……51
4.4.2 内存池类型 ……52
4.4.3 创建顶点缓存 ……52
4.4.4 渲染顶点缓存图形 ……54
4.5 使用索引缓存绘制图形 ……55
4.5.1 创建顶点缓存和索引缓存 ……56
4.5.2 使用索引缓存绘制图形 ……57
4.6 二维文本绘制 ……58
4.6.1 创建ID3DXFont对象 ……58
4.6.2 使用ID3DXFont接口绘制二维文本 ……59
4.6.3 帧速率计算 ……60
4.7 渲染状态 ……60
4.7.1 着色模式 ……61
4.7.2 多边形填充模式 ……61
*4.8 图形反锯齿 ……62
4.8.1 查询设备是否支持多重采样 …62
4.8.2 创建使用多重采样的Direct3D设备对象 ……62
4.8.3 启用多重采样 ……63
4.9 小结 ……63
练习……63
第5章 顶点坐标变换 ……64
5.1 顶点坐标变换和光照流水线概述 ……64
5.2 矩阵类型 ……65
5.2.1 D3DMATRIX ……65
5.2.2 D3DXMATRIX……65
5.2.3 D3DXMATRIXA16 ……67
5.3 坐标变换 ……67
5.3.1 世界变换 ……67
5.3.2 观察变换 ……70
5.3.3 投影变换 ……71
5.3.4 视区变换 ……72
5.4 坐标变换示例程序 ……72
5.5 小结 ……74
练习……75
第6章 光照 ……76
6.1 光照计算模型 ……76
6.1.1 环境光 ……76
6.1.2 漫反射光 ……76
6.1.3 镜面反射光 ……77
6.1.4 自发光 ……77
6.2 光源 ……78
6.2.1 光源类型 ……78
6.2.2 光源属性 ……79
6.2.3 设置与启用光源 ……80
6.2.4 对光源的几点说明 ……81
6.3 材质 ……81
6.4 光照和材质示例程序 ……82
6.5 对光照和材质的几点说明 ……84
6.6 小结 ……84
练习……84
第7章 纹理映射 ……85
7.1 纹理元素与纹理坐标 ……85
7.2 创建纹理对象 ……85
7.3 纹理过滤方式 ……86
7.3.1 最近点采样 ……86
7.3.2 线性纹理过滤 ……86
7.3.3 各向异性纹理过滤 ……87
7.3.4 多级渐进纹理过滤 ……87
7.4 纹理寻址模式 ……88
7.4.1 重叠纹理寻址模式 ……89
7.4.2 镜像纹理寻址模式 ……89
7.4.3 夹取纹理寻址模式 ……90
7.4.4 边框颜色纹理寻址模式 ……90
7.5 纹理映射示例程序 ……91
7.6 纹理阶段混合状态 ……92
7.6.1 纹理阶段混合状态设置 ……92
7.6.2 纹理阶段混合状态示例程序 …93
7.7 多层纹理映射 ……94
7.8 小结 ……96
练习……97
第8章 Direct3D渲染技巧 ……98
8.1 Alpha混合 ……98
8.1.1 Alpha混合原理 ……98
8.1.2 Alpha混合系数 ……99

8.1.3 Alpha混合示例 ……99
8.2 雾化效果 ……101
8.2.1 雾化效果实现原理 ……101
8.2.2 基于范围的雾化 ……102
8.2.3 雾化效果示例程序 ……103
8.2.4 对雾化效果的几点说明 ……104
*8.3 模板测试 ……104
8.3.1 模板测试过程 ……104
8.3.2 模板缓存 ……105
8.3.3 模板测试设置 ……106
8.3.4 平面阴影 ……107
8.4 小结 ……110
练习 ……110
第9章 网格模型 ……111
9.1 网格模型基础 ……111
9.1.1 几何信息 ……111
9.1.2 网格子集和属性缓存 ……112
9.1.3 邻接信息 ……113
9.1.4 绘制网格模型 ……113
9.1.5 克隆网格模型 ……114
9.1.6 创建网格模型 ……114
9.2 简单几何体和三维文本网格模型 ……115
9.2.1 创建简单几何体网格模型 ……115
9.2.2 创建三维文本网格模型 ……116
9.3 文件网格模型 ……117
9.3.1 模型文件格式 ……117
9.3.2 使用.X文件网格模型 ……118
9.3.3 对网格模型的几点说明 ……121
9.4 网格模型优化 ……122
9.4.1 优化函数和优化方式 ……122
9.4.2 属性表 ……123
9.4.3 其他网格模型处理函数 ……124
9.4.4 网格优化示例程序 ……125
9.5 文件网格模型类 ……126
9.5.1 CXFileMesh类的定义 ……126
9.5.2 CXFileMesh的实现 ……127
9.5.3 使用CXFileMesh类渲染文件网格模型 ……129
9.6 物体空间状态的控制 ……130
*9.7 蒙皮骨骼动画网格模型 ……132
9.7.1 骨骼动画基本原理 ……133
9.7.2 图形混合技术 ……134
9.7.3 蒙皮骨骼动画网格模型类介绍 ……136
9.8 小结 ……141
练习 ……141
第10章 点精灵与粒子系统 ……143
10.1 粒子系统基本原理 ……143
10.2 粒子和点精灵 ……144
10.2.1 点精灵顶点结构和顶点格式 ……144
10.2.2 点精灵的大小 ……144
10.2.3 点精灵的纹理坐标 ……145
10.2.4 点精灵的渲染 ……145
10.3 粒子系统的具体实现 ……146
10.3.1 粒子结构与顶点结构 ……146
10.3.2 粒子系统类的定义 ……146
10.3.3 粒子系统的创建和销毁 ……147
10.3.4 粒子系统的更新 ……148
10.3.5 粒子系统的渲染 ……150
10.3.6 粒子系统示例程序 ……152
10.4 雪景模拟 ……154
10.4.1 雪花粒子结构 ……154
10.4.2 雪花粒子运动的控制 ……154
10.4.3 渲染雪花粒子 ……155
10.5 小结 ……156
练习 ……156
第11章 基本虚拟场景构造 ……157
11.1 三维地形模拟 ……157
11.1.1 基本地形网格的构造 ……157
11.1.2 高度图与高程数据 ……161
11.1.3 地形上任意点高度的计算 ……162
11.1.4 法线的计算 ……163
11.1.5 地形模拟示例程序 ……164
11.2 天空绘制 ……165
11.2.1 矩形天空 ……165
11.2.2 天空盒 ……166
11.2.3 球形天空 ……167
11.3 虚拟摄像机与虚拟场景漫游 ……168
11.3.1 虚拟场景漫游基本原理 ……168
11.3.2 CCamera类 ……169
11.3.3 虚拟场景漫游的实现 ……172
11.4 小结 ……173
练习 ……174
第12章 HLSL高级着色语言 ……175
12.1 数据类型 ……175
12.1.1 标准数据类型 ……175
12.1.2 向量 ……176
12.1.3 矩阵 ……177

12.1.4 复杂数据类型 ……179
12.1.5 变量的前缀 ……181
12.2 运算符和语句 ……181
12.2.1 运算符 ……181
12.2.2 类型转换 ……182
12.2.3 语句 ……182
12.3 函数 ……183
12.3.1 自定义函数 ……183
12.3.2 内置函数 ……183
12.4 语义 ……183
12.5 编写HLSL着色器 ……185
12.5.1 添加文件 ……185
12.5.2 编写着色器代码 ……185
12.5.3 编译着色器代码 ……186
12.6 小结 ……187
练习 ……187
第13章 着色器和效果 ……188
13.1 顶点着色器 ……188
13.1.1 顶点着色器介绍 ……188
13.1.2 顶点声明 ……189
13.1.3 使用HLSL顶点着色器的基本步骤 ……190
13.1.4 使用顶点着色器实现基本光照模型 ……194
13.2 像素着色器 ……199
13.2.1 像素着色器介绍 ……199
13.2.2 使用像素着色器的基本步骤…199
13.2.3 使用像素着色器实现纯亮度显示 ……202
13.3 效果 ……203
13.3.1 效果介绍 ……203
13.3.2 创建效果 ……203
13.3.3 使用效果 ……205
13.3.4 多手法效果 ……206
*13.3.5 使用效果管理状态 ……209
13.4 小结 ……212
练习 ……212
第14章 DirectX Audio音频编程 ……214
14.1 DirectX Audio介绍 ……214
14.1.1 DirectX Audio技术架构 ……214
14.1.2 DirectX Audio支持的音频格式 ……215
14.2 XACT ……216
14.2.1 XACT介绍 ……216
14.2.2 使用XACT GUI制作声音 …218
14.2.3 XACT初始化 ……219
14.2.4 使用XACT播放声音 ……221
14.2.5 使用XACT播放流声音 ……225
14.2.6 代码驱动的XACT API介绍 …229
14.3 XAudio2 ……229
14.3.1 XAudio2介绍 ……229
14.3.2 XAudio2初始化 ……231
*14.3.3 使用XAudio2加载音频数据 ……232
14.3.4 使用XAudio2播放声音 ……235
14.4 小结 ……237
练习 ……237
第15章 DirectInput与场景交互 ……239
15.1 DirectInput初始化 ……239
15.1.1 创建DirectInput对象 ……239
15.1.2 枚举输入设备 ……240
15.1.3 创建DirectInput设备对象 …241
15.1.4 设置设备对象 ……241
15.1.5 获取输入设备的访问权 ……243
15.2 键盘输入处理 ……244
15.3 鼠标输入处理 ……246
15.4 游戏杆输入处理 ……249
*15.5 拾取 ……251
15.5.1 拾取射线的计算 ……252
15.5.2 拾取射线的变换 ……253
15.5.3 拾取射线与物体交点的计算…254
15.5.4 拾取示例程序 ……254
15.6 小结 ……255
练习 ……255
*第16章 DXUT程序框架 ……256
16.1 DXUT程序框架剖析 ……256
16.1.1 DXUT框架文件组成和功能 256
16.1.2 DXUT框架回调函数 ……257
16.2 使用DXUT框架 ……260
16.2.1 实现并设置回调函数 ……260
16.2.2 初始化DXUT ……261
16.2.3 其他DXUT设置 ……261
16.2.4 创建窗口 ……262
16.2.5 创建Direct3D设备 ……262
16.3 使用DXUT绘制文本 ……263
16.4 使用DXUT控件 ……265
16.4.1 初始化对话框 ……265
16.4.2 渲染控件 ……266

此为试读，需要完整PDF请访问：www.ertongbook.com

16.4.3 处理控件事件 ……267
16.4.4 释放对话框 ……267
16.5 综合实例 ……268
16.5.1 生成程序框架 ……269
16.5.2 构造基本场景 ……269
16.5.3 添加背景音乐 ……271
16.5.4 添加输入处理 ……271
16.5.5 实现场景漫游 ……272
16.6 小结 ……273
练习 ……274
附录A Direct3D 10与Direct3D 11介绍 ……275
参考文献 ……282

第1章 DirectX与游戏编程简介

本章首先介绍DirectX的基本概念和技术背景，使读者对DirectX有一个大致的理解。然后介绍DirectX的功能组件、DirectX与游戏开发、DirectX的安装与配置等内容，使读者能够从整体上了解DirectX及其相关内容，为后续学习奠定基础。

1.1 DirectX介绍

DirectX最初是为了弥补Windows 3.1对图形、声音处理能力的不足，作为一个附加工具包，用于在Windows平台上开发游戏。现在DirectX已经成为Windows系统自身的重要组成部分，以支持各种现代显卡，并且已经成为对整个多媒体系统的各方面都有决定性影响的开发接口。

对于通常所说的DirectX实际上有两种不同的含义：一种含义是指DirectX SDK（俗称“DirectX开发工具包”）或称为DirectX API（DirectX应用程序编程接口），它是微软公司提供的一套用于开发高性能多媒体程序的编程接口；另外一种含义是指DirectX Runtime（DirectX运行库），它是一组动态链接库，是运行使用DirectX SDK开发的程序所需要的动态库。在安装游戏程序时，经常会弹出一个提示框，询问用户是否安装新版本的DirectX，这时所提到的DirectX就是指DirectX Runtime。对于开发人员来说，DirectX主要是指DirectX SDK，本书中提到DirectX时，如无特殊说明也是指DirectX SDK。另外，对于初次接触DirectX的用户来说，还应注意区分DirectX与Director，Director是微软公司开发的多媒体课件制作软件。因为二者的发音相似，所以初学者很容易混淆。

DirectX并不是一个单纯的图形API，它主要包含DirectX Graphics、DirectX Audio、DirectX Input三大部分，每一部分又包含多个功能组件，共同构成了一套完整的多媒体开发接口。这套开发接口主要是以COM组件的形式提供给程序员使用的。其中DirectX Graphics专门负责图形（以及动画）处理、DirectX Audio提供了对高质量声音效果的支持，DirectX Input提供了对各种输入设备的支持。有关DirectX各功能组件的具体功能，在本章后面还会专门介绍。

在DirectX SDK提供的众多功能组件中，Direct3D是专门负责图形开发的一个组件，也是DirectX中最重要、最复杂的组件。随着Direct3D版本的不断更新，其功能越来越强大，图形硬件对Direct3D的支持也越来越好。往往在新版本的Direct3D刚刚发布不久，硬件厂商就会生产出支持最新版本Direct3D功能的显卡。目前Direct3D已经成为三维图形开发领域事实上的标准，甚至在最新的Windows Vista、Windows 7操作系统中将Direct3D作为操作系统的图形引擎，而放弃了过去一直使用的GDI/GDI+。

*1.2 OpenGL、XNA与WPF

严格来说，本节内容不属于本书的范畴，但是这些内容与DirectX密切相关，而且也是目前流行的图形以及界面编程技术。作为技术背景，了解OpenGL、XNA以及WPF是什么，以及它们与DirectX之间的关系，可帮助加深对DirectX的理解。

1.2.1 OpenGL

OpenGL（Open Graphics Library，开放性图形库）源于SGI公司为其图形工作站开发的IRIS GL，在跨平台的移植过程中发展成为OpenGL，是一套和Direct3D类似的三维图形开发接口。SGI公司在1992年7月发布了OpenGL 1.0版，OpenGL则成为显卡工业的一项特定标准。接下来OpenGL由成立于1992年的独立财团OpenGL Architecture Review Board（ARB）所控管。在1995年12月ARB推出了OpenGL 1.1版本，1999年推出了1.2.1版本，现在OpenGL的最新版本是2.0。

目前，三维图形的底层开发基本上都是使用OpenGL或Direct3D，也就是说在三维图形以及游戏开发领域，OpenGL是Direct3D唯一的竞争对手。两者在功能上非常类似，都是用于三维图形开发的编程接口。它们之间的区别主要有以下几个方面。

- 开发平台方面。OpenGL提供了跨平台支持，可以运用于当前各种流行的操作系统之上，如Windows、Linux、UNIX、Mac OS等。而Direct3D只能用于Windows操作系统。这是OpenGL相对于Direct3D最主要的优点。
- 应用领域方面。最初OpenGL主要应用于专业图形领域，而Direct3D主要应用于Windows平台下的游戏开发。不过目前这种区别已经不明显了，Direct3D也逐步涉足专业图形领域，最近几年出现的许多专业图形开发平台，例如Virtools，在底层就是基于Direct3D开发的。而在以前，专业图形开发平台，例如Vega、WTK等，在底层基本上都是基于OpenGL开发的。
- 图形质量和运行速度方面。最初OpenGL主要关注的是图形质量，而Direct3D主要关注运行速度。为此Microsoft和显卡厂家紧密合作，为Direct3D提供复杂纹理映射、特殊效果（例如半透明）以及三维图形所需的硬件加速功能。不过，自从2002年发布了Direct3D 9.0以来，Direct3D在图形质量方面有了明显的提高，其图形质量并不比OpenGL差，而最近推出的Direct3D 10和DirectX 11，在图形质量方面又有了长足的进步。
- 接口形式方面。Direct3D是以COM接口的形式提供的，而OpenGL是以动态链接库的形式提供的，采用原始C语言函数的调用方式。所以OpenGL相对于Direct3D，学习起来要相对容易些，但是在具体使用时则不如Direct3D方便。

前面提到过，Direct3D仅仅是DirectX中专门负责图形部分的功能组件。除了Direct3D之外，还有负责音频开发的DirectX Audio，以及用于处理输入的DirectX Input。OpenGL也有一套专门用于音频开发的Open AL编程接口。同样，Open AL相对于DirectX Audio其主要优点是可以跨平台，但是在音频质量方面要比DirectX Audio逊色得多。

1.2.2 XNA

微软在2000年6月宣布了.NET战略，.NET是微软公司的新战略，所有微软公司的产品都将围绕这个战略开发。.NET平台对早期的开发平台进行了显著增强，提供了一种全新的软件

开发模式和组件标准。自从2002年微软发布了第一个版本的.NET Framework（随同Visual Studio .NET一同发布）以来，.NET便风靡一时，成为最流行的开发平台之一。目前许多商业二次开发工具包，逐步从原来的COM开发标准转而采用.NET开发标准。

微软为了迎合自己的.NET战略，一直在努力将DirectX移植到.NET框架中。随着DirectX 9的发布，在DirectX中加入了Managed DirectX 1.0。顾名思义，Managed DirectX是.NET平台上的DirectX，它封装了DirectX中的大部分功能，几乎实现了函数接口的一一对应，以便开发人员在.NET平台上编写高性能多媒体应用程序。

实际上微软在开发DirectX 8.1的后期，就有一个被称之为DirectX for Visual Basic的项目，当.NET的第一个beta版本出现后，这个项目变成了DirectX .NET，也就是Managed DirectX的雏形。

微软在2005年10月发布的DirectX SDK中，加入了Managed DirectX 2.0的第一个beta版本，但是在这之后就终止了Managed DirectX的一切计划。后来微软宣布放弃Managed DirectX，取而代之的是XNA。

XNA是基于.NET的、专用于游戏开发的API。Managed DirectX是对DirectX的一个.NET包装，实质上是调用DirectX API。XNA虽然也是基于DirectX实现的，但是没有像Managed DirectX那样直接调用DirectX API，XNA对DirectX提供了更高层次的封装。同样XNA也不单纯是一套图形API，它支持图形、声音、输入以及网络等多方面的功能，其中三维图形和动画处理也是XNA最重要的功能。

由于在游戏开发领域，一直主要是采用C/C++作为开发语言，所以转为使用XNA进行游戏开发需要一个过程，再加上.NET程序在性能上多少要比C/C++程序逊色一些，所以从目前来看这个过程还是比较缓慢。不过鉴于.NET开发的诸多优点，采用XNA开发高性能多媒体程序也是一个不错的选择。由于XNA在底层是基于DirectX开发的，因此在学习使用XNA之前，掌握DirectX的基础知识还是很有必要的，就像学习.NET开发需要掌握一定的Windows API基础一样。目前XNA的最新版本是3.1，其底层是基于DirectX 9开发的。

1.2.3 WPF

相对于OpenGL和XNA，WPF与本书的主题偏离得更远。鉴于WPF是未来用户界面开发的主流技术，并且在底层也是基于DirectX开发的，所以了解WPF的基本情况，对于理解DirectX这一所谓的图形标准还是很有益处的。

要介绍WPF，还得先从GDI/GDI+说起。GDI（Graphics Device Interface，图形设备接口）是早期Windows的图形子系统，它负责在视频显示器和打印机上显示文本和图形。GDI是Windows非常重要的组成部分，不但程序员为Windows编写的应用程序在显示视觉信息时需要使用GDI，就连Windows操作系统本身也是使用GDI来显示用户界面对象的，例如菜单、滚动条、图标和鼠标光标等。

虽然GDI提供了非常强大的图形功能，而且使用GDI在窗口上绘制图形也比较简单，但是GDI只是一个静态的显示系统，只有有限的动画支持，此外GDI只支持二维图形开发。所以GDI不适合开发高性能多媒体应用程序，例如游戏程序，于是微软以一个附加开发工具包的形式，提供了DirectX。

之后Windows平台上的图形开发，基本上沿着两条道路向前发展，一条是适用于二维静态图形（主要是用户界面）开发的GDI，另外一条便是适应于三维图形和动画开发的Direct3D。

其中GDI逐步发展为GDI+，GDI+是早期Windows系统图形引擎GDI的继任者，是Windows XP和Windows Server 2003操作系统的图形引擎。目前常用的应用程序开发平台，无论是基于.NET的Windows窗体，还是以前Visual Basic中的窗体或基于C++代码的MFC，以及Delphi和C++ Builder中的窗体，在底层都是使用GDI/GDI+负责用户界面的显示。

另一方面Direct3D的发展要比GDI/GDI+的发展迅速得多，版本不断升级，而且极大地推动了图形硬件的发展。而图形硬件的快速发展反过来又在一定程度上推动了图形技术的发展，为普及本来只局限于高性能图形领域的Direct3D提供了硬件基础。再加上丰富视觉界面体验需求的快速增加，使得本来作为附加部分的Direct3D却逐渐成为主流。对于Windows操作系统本身的图形引擎，从Windows Vista开始在底层也是采用Direct3D，而放弃了20多年来一直采用的GDI/GDI+。但是Direct3D正如最初设计它的目的，主要面向高性能多媒体应用程序。因为它本来就很复杂，所以Direct3D几乎无法用于开发传统类型的Windows应用程序（例如商业软件）。毕竟，使用底层的Direct3D开发接口构建由各种控件组成的用户界面，确实不是一件很容易的事情，于是WPF便应运而生。

WPF（Windows Presentation Foundation，Windows呈现基础），是Windows Vista和Windows 7的图形子系统，提供了整合图形用户界面的完整方案，包括二维与三维图形、文档以及数字媒体。在WPF中，底层的图形技术不再是GDI/GDI+，而是Direct3D。不管创建哪种用户界面，WPF应用程序在底层都是使用Direct3D。这意味着，不管是设计复杂的三维图形（这是Direct3D的特长），还是仅仅绘制几个按钮以及简单的文本，所有绘图工作都是通过Direct3D流水线（pipeline）完成的。因此，即使是最普通的商业应用程序，也能够使用丰富的效果，例如半透明和反走样。使用WPF还可以受益于硬件加速，为提高性能Direct3D程序通常将尽可能多的工作交给GPU（图形处理单元，显卡专用的处理器）进行处理，为此需要进行许多复杂的处理，而WPF简化了这一处理过程。

图1.1显示了Windows图形技术的发展变化情况：最初是GDI/GDI+和Direct3D并列发展；现在逐步发展为以Direct3D为基础的WPF取代了原来GDI/GDI+负责的领域，而Direct3D的传统应用领域，除了Direct3D本身之外，又出现了XNA。当然WPF和XNA都是基于.NET的图形开发技术，而Direct3D是基于底层API的本机开发技术。总之Direct3D现在不再是一个用于特殊领域的附加开发工具包，而已经成为整个Windows系统的图形基础。

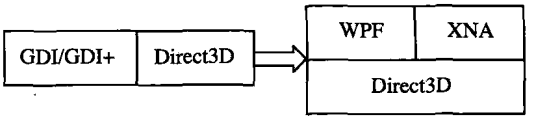

图1.1 Windows图形技术的发展变化

提示：WPF作为Windows用户界面的开发技术，取代基于GDI/GDI+的用户界面开发技术也需要一个过程。基于GDI/GDI+的图形界面开发技术相对于WPF更加成熟，并且许多高级特征还没有完全集成到WPF工具包中。

1.3 DirectX功能组件

DirectX SDK是一套丰富的开发接口，包含不同的功能组件，了解DirectX SDK的各个功能组件，可以从整体上把握DirectX SDK提供的功能。随着DirectX SDK版本和功能的不断升级，DirectX的功能组件也一直在不断变化。现在DirectX的功能组件被划分三大部分：DirectX Graphics、DirectX Audio和DirectX Input。

提示：从DirectX 9.0开始，由于各个功能组件的版本更新不再一致，而且有些功能组件，例如Direct3D，还出现了多个版本共存的情况。所以DirectX的版本现在也不再以数字进行标识，而是使用发布日期进行标识，例如本书使用的是2009年8月发布的DirectX SDK，即DirectX SDK（August 2009）。

1.3.1 DirectX Graphics

DirectX Graphics是DirectX SDK中负责图形开发的部分。在DirectX的早期版本中，图形处理部分包括负责二维图形开发的DirectDraw和负责三维图形开发的Direct3D。从DirectX 8.0版之后，二维图形和三维图形统一由Direct3D处理，而不再单独提供DirectDraw，同时将负责图形处理的各个组件（包括一些辅助组件）统一称为DirectX Graphics。由于Direct3D是DirectX Graphic中最主要的组件，并且也一直是DirectX中最复杂的组成部分，以至于经常将Direct3D的版本作为DirectX的版本，例如有时将带有Direct3D 10的DirectX版本称为DirectX 10，甚至有时Direct3D也成了DirectX的代名词。

在DirectX SDK（December 2006）版本中发布了Direct3D 10，Direct3D 10是一个全新的产品，而不是Direct3D 9的升级版本。Direct3D 10和Direct3D 9在底层技术和架构上完全不同，于是第一次出现了两个版本并存的情况，其中Direct3D 10只能在Windows Vista以及最新的Windows 7操作系统上运行。在DirectX SDK（March 2008）版本中，Direct3D 10升级到了Direct3D 10.1。

在DirectX SDK（August 2009）版本中微软又发布了Direct3D 11，Direct3D 11在引入新技术的同时保留了Direct3D 10.1的全部特性。换句话说，Direct3D 11实际上是Direct3D 10.1的扩展集，所有Direct3D 10.1硬件所遵循的API，对于Direct3D 11同样适用。由于Direct3D 11是Direct3D 10.1的扩展而不是升级，所以Direct3D 11不能完全取代Direct3D 10.1，而且Direct3D 11也只能运行于Windows Vista以及Windows 7操作系统上。于是目前是Direct3D 9、Direct3D 10/10.1和Direct3D 11三个版本共存。

由于目前Windows XP仍然是主流的操作系统，而且Direct3D 9不仅能运行于Windows XP以及Windows 2000系统之上，而且在Windows Vista和Windows 7操作系统之上也可以运行。加之对于熟悉Direct3D 9的开发人员来说，学习Direct3D 10和Direct3D 11几乎是重新开始。所以Direct3D 9仍然是目前Direct3D的主流版本。显然只有等到Windows Vista以及Windows 7成为主流的操作系统之后，Direct3D 10和Direct3D 11才会逐步代替Direct3D 9成为Direct3D的主流版本。鉴于目前的这种现状，本书以Direct3D 9为主，同时考虑到以后的发展，在附录中对Direct3D 10和Direct3D 11进行了简要介绍。

1.3.2 DirectX Audio

DirectX Audio是专门负责音频开发的部分。其中包含用于控制Xbox 360和Windows平台上音频开发的API函数，并且提供了相应的工具，用于生成和编码这些API函数可以使用的音频数据。

在DirectX音频开发领域，最著名的开发接口莫过于DirectSound，它是在DirectX中最早提供的音频编程组件。DirectSound提供了对高质量声音效果的支持，除了能够播放WAV格式的声音和处理混音之外，还可以为程序添加极具真实感的三维声音效果。此外，DirectSound还提供了从话筒或其他输入设备捕获WAV声音的功能。

此为试读，需要完整PDF请访问：www.ertongbook.com