

《DirectX游戏编程》

图书基本信息

书名：《DirectX游戏编程》

13位ISBN编号：9787302215936

10位ISBN编号：7302215936

出版时间：2010-1

出版社：周国庆、陈洪、冯人果 清华大学出版社 (2010-01出版)

作者：周国庆,陈洪,冯人果

页数：365

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

前言

这是一本讲述如何使用DirectX 9.0编写交互式三维计算机图形程序的书，重点在游戏开发方面。将讲述关于Direct3D的基础知识，使读者能够进一步地学习和使用更高级的技术。当读者看完这本书的时候，对DirectX就会有一个大体的了解。从程序员的角度看，DirectX是一个用于开发Windows平台下多媒体应用程序的API(Application Programming Interfaces)函数的集合。本书重点关注DirectX的一个重要子集——Direct3D。就像名字所描述的，Direct3D是一个用于开发三维应用程序的API的集合。当然本书同时也涉及DirectX的其他内容。本书主要分为4个部分：第1部分主要介绍贯穿于本书的基本数学概念；第2部分将会介绍基本的3D技术，例如光照、纹理、Alpha混合以及模板缓冲区；第3部分主要讲述如何通过Direct3D去实现多种有趣的技术和应用程序，例如图元选取(Picking)、地形渲染(Terrain Render)、粒子系统(Particle System)、通用虚拟摄像机(Camera)以及如何加载和渲染3D模型(.x文件)；第4部分的主题是Vertex Shader、Pixel Shader以及特效框架和新的(在DirectX 9中首次出现)高级Shader语言(HLSL)。当前和未来的三维计算机游戏会更多地运用Shader技术，所以这是一本与现行图形编程密切相关的书。

1.对读者的要求 本书属于基础性的图书，但是这并不表示本书适合于没有任何编程经验的读者。本书要求读者了解代数学、几何学的知识，熟悉游戏程序开发环境(例如Visual C++)，掌握C++语言以及链表和数组等基本的数据结构知识，如果读者对Windows程序设计具有一定的了解，将更有助于阅读本书，不过这不是必需的。读者可以通过附录A对Windows程序设计有一个简单的认识。

内容概要

《DirectX游戏编程》主要讲述了利用Direct3D进行图形程序开发的核心技术，从三维向量开始，由浅入深，逐步讲解3D图形编程中的矩阵变换相关知识、投影知识，及视口处理、裁剪处理、深度缓存的应用技术。在高级绘图技术中进一步讲解纹理处理、透明及混合技术的应用、模板的应用、光照及阴影技术、网格的应用、利用网格技术处理模型显示、在Direct3D程序中显示文字、粒子系统的创建及应用。书的后半部分讲解游戏设计的高级特性，如摄像机封装技术、地形显示技术、场景中的模型捡取。最后讲解GPU编程相关的知识，让用户了解高级着色的相关内容。

《DirectX游戏编程》每章都以简单的实例来辅助对知识点的讲解，力求深入浅出，知识由浅入深，难度是逐渐递增的，另外《DirectX游戏编程》的章节之间没有过分紧密的程序连接，所以有一定基础的人员，可以选择需要的章节单独阅读。

《DirectX游戏编程》面向的群体为高校数字媒体艺术程序设计专业的学生，及自考学生，或有一定编程基础的专业图形编程人员。

书籍目录

第1部分 必备的数学知识第1章 基本的数学概念 31.1 三维空间中的向量 31.1.1 向量的相等 61.1.2 向量模的计算 71.1.3 向量归一化 81.1.4 向量加法 81.1.5 向量减法 91.1.6 数乘 101.1.7 点乘 101.1.8 叉乘 111.2 矩阵 121.2.1 矩阵的相等、矩阵数乘和矩阵加法 131.2.2 矩阵乘法 131.2.3 单位矩阵 151.2.4 逆矩阵 151.2.5 矩阵的转置 161.2.6 D3DX中的矩阵 161.2.7 基本变换 191.2.8 平移变换 201.2.9 旋转矩阵 211.2.10 缩放矩阵 231.2.11 组合变换 231.2.12 一些用于向量变换的函数 251.3 平面 251.3.1 D3DX中的平面 261.3.2 点和平面之间的关系 271.3.3 构建平面 271.3.4 标准化一个平面 281.3.5 对平面进行变换操作 291.3.6 求到一个指定点距离最短的平面上的点 291.4 射线 301.4.1 射线的表达式 301.4.2 射线与平面的相交 311.5 小结 31 第2部分 Direct3D基础知识第2章 Direct3D初始化 352.1 Direct3D体系结构 352.1.1 REF设备 362.1.2 D3D设备类型 362.2 组件对象模型 372.3 预备知识 372.3.1 表面缓冲区(Surface) 372.3.2 重采样 392.3.3 像素格式 402.3.4 存储池 402.3.5 交换链和页面翻转 412.3.6 深度缓冲区 422.3.7 顶点处理 432.3.8 设备性能 432.4 初始化Direct3D 442.4.1 获得IDirect3D9接口 452.4.2 确认硬件顶点处理能力 452.4.3 填充D3DPRESENT_PARAMETERS结构 462.4.4 创建一个IDirect3DDevice9接口 482.5 示例：初始化Direct3D 492.5.1 d3dUtility.h/cpp文件 502.5.2 程序框架 522.5.3 D3D初始化示例 522.6 小结 55第3章 渲染管道 573.1 描述的模型 573.1.1 顶点格式 583.1.2 三角形 593.1.3 索引 603.2 虚拟摄像机 603.3 渲染管道 613.3.1 本地空间 623.3.2 世界空间 623.3.3 视图空间 633.3.4 背面裁剪 653.3.5 光照 663.3.6 裁剪 663.3.7 投影 673.3.8 视口变换 693.3.9 光栅化 703.4 小结 70第4章 在Direct3D中绘制图形 714.1 顶点/索引缓冲区 714.1.1 创建顶点缓冲区和索引缓冲区 714.1.2 访问缓冲区内内存 744.1.3 获得顶点缓冲区以及索引缓冲区的相关信息 754.2 渲染状态 764.3 渲染前的准备工作 774.4 使用顶点/索引缓冲区进行渲染 784.4.1 IDirect3DDevice9::DrawPrimitive 784.4.2 IDirect3DDevice9::DrawIndexedPrimitive 784.4.3 Begin/End Scene 804.5 D3DX自带的几何物体 804.6 示例：三角形、立方体、茶壶、D3DXCreate* 814.7 小结 85第5章 颜色 875.1 颜色的描述方法 875.2 顶点颜色 905.3 着色处理 905.4 示例：带颜色的三角形 915.5 小结 93第6章 光照 956.1 光的组成 956.2 材质 966.3 顶点法线 986.4 光源 1006.5 示例：光照 1046.6 其他例子 1066.7 小结 107第7章 纹理 1097.1 纹理坐标 1097.2 创建并应用纹理 1117.3 纹理过滤 1127.4 多纹理映射 1137.4.1 多纹理映射过滤 1137.4.2 在Direct3D中使用多纹理映射 1147.5 纹理坐标寻址模式 1147.6 程序示例：带纹理的四边形 1167.7 小结 118第8章 混合技术 1198.1 混合方程 1198.2 混合因子 1218.3 透明 1218.3.1 Alpha通道 1228.3.2 指定源像素的Alpha值 1228.4 用DirectX的纹理工具创建Alpha通道 1238.5 示例：透明效果 1248.6 小结 127第9章 模板缓冲区 1299.1 使用模板缓冲区 1309.1.1 获取模板缓冲区 1318.1.2 模板测试 1319.1.3 模板测试的控制 1319.1.4 更新模板缓冲区 1339.1.5 模板写入掩码 1349.2 示例：镜子 1349.2.1 镜面反射的相关数学知识 1349.2.2 镜面效果的实现概述 1369.2.3 代码和解释 1379.3 示例：平面阴影 1409.3.1 平行光阴影 1419.3.2 点光源阴影 1429.3.3 阴影矩阵 1429.3.4 用模板缓冲区来防止二次混合 1439.3.5 代码和解释 1449.4 小结 146 第3部分 Direct3D的应用第10章 文字处理 14910.1 ID3DXFont、ID3DXFont接口 14910.1.1 创建一个ID3DXFont接口 14910.1.2 文字输出 15010.1.3 计算帧速(FPS) 15110.2 CD3DFont、CD3DFont类 15210.2.1 构造CD3DFont对象 15210.2.2 DrawText输出文字 15310.2.3 销毁CD3DFont对象 15310.3 三维网格(3D Mesh)在文字显示中的应用 15410.3.1 D3DXCreateText方法 15410.4 小结 155第11章 网格(1) 15711.1 几何信息 15711.2 网格子集(Subset)和属性缓冲区 15811.3 网格的渲染 16011.4 网格优化 16011.5 网格属性表 16211.6 邻接信息 16411.7 网格的克隆(Cloning) 16511.8 网格的创建 (D3DXCreateMeshFVF) 16611.9 示例：创建并渲染一个网格 16811.10 小结 173第12章 网格(2) 17512.1 ID3DXBuffer接口 17512.2 .X文件 17612.2.1 读取.X文件 17712.2.2 .X文件中的材质 17812.2.3 .X文件实例 17812.2.4 生成顶点法线 18112.3 渐进式网格 18212.3.1 生成渐进式网格 18312.3.2 顶点属性权重 18412.3.3 ID3DXPMesh接口的相关方法 18512.3.4 示例：渐进式网格 18612.4 包围体 18912.4.1 新的常量 19112.4.2 各种不同的包围体 19112.4.3 包围体示例 19212.5 小结 194第13章 构建一个灵活的摄像机类 19513.1 设计摄像机 19513.2 摄像机类的实现细节 19713.2.1 计算视图矩阵 19713.2.2 绕任意轴的旋转 20013.2.3 俯仰、偏转和翻滚 20013.2.4 行走、平移和飞行 20313.3 示例：摄像机 20413.4 小结 206第14章 基本地形渲染 20714.1 高度图 20814.1.1 高度图的创建 20814.1.2 读取RAW文件 20914.1.3 高度图的访问和修改 21014.2 生成地形几何数据 21014.2.1 计算顶点 21214.2.2 计算索引——定义三角形 21414.3 纹理 21614.3.1 用程序生成纹理数据 21614.4 光照 21814.4.1 概述 21914.4.2 计算方格的阴影 21914.4.3 计算地形阴影 22114.5 在地形上“行走” 22114.6 Terrain示例程序 22514.7 一些改进措施 22614.8 小结 227第15章 粒子系统 22915.1 粒子和点精灵 22915.1.1 结构格式 23015.1.2 Point Sprite渲染

染状态 23015.1.3 粒子及其属性 23215.2 粒子系统组件 23315.2.1 粒子系统的渲染 23815.2.2 粒子的随机性
24115.3 实现三个具体的粒子系统：雪景、焰火和粒子枪 24215.3.1 示例：雪景 24215.3.2 示例：焰火
24415.3.3 示例：粒子枪 24615.4 小结 248第16章 捡取 24916.1 屏幕到投影窗口的变换 25116.2 计算Picking
射线 25216.3 变换射线 25316.4 射线与物体的相交检测 25416.5 示例：Picking 25616.6 小结 257第4部分
Shader和特效第17章 高级Shader语言的介绍 26117.1 使用HLSL编写Shader程序 26217.1.1 全局变量
26317.1.2 输入输出结构 26417.1.3 入口点函数 26517.2 编译一个HLSL Shader程序 26617.2.1 常量表
26617.2.2 编译HLSL Shader程序 27017.3 变量类型 27217.3.1 标量类型 27217.3.2 向量类型 27217.3.3 矩阵类
型 27317.3.4 数组 27517.3.5 结构 27517.3.6 typedef关键字 27617.3.7 变量前缀 27617.4 关键字、语句和类型
转换 27717.4.1 关键字 27717.4.2 基本的程序流程 27717.4.3 类型转换 27817.5 运算符 27917.6 用户自定义函
数 28017.7 内建函数 28217.8 小结 284第18章 Vertex Shader的介绍 28518.1 顶点声明(Vertex Declaration)
28618.1.1 顶点声明的描述 28618.1.2 创建顶点声明 28918.1.3 启用一个顶点声明 28918.2 顶点数据的用途
29018.3 使用Vertex Shader的步骤 29218.3.1 编写与编译 Vertex Shader 29218.3.2 创建Vertex Shader 29218.3.3
设置Vertex Shader 29318.3.4 销毁Vertex Shader 29318.4 示例：散射光 29318.5 示例：卡通渲染 (Cartoon
Rendering) 29918.5.1 卡通着色 (Cartoon Shading) 30018.5.2 卡通着色的Vertex Shader 代码 30118.5.3 勾画出
物体的轮廓 30318.6 小结 308第19章 Pixel Shader入门 30919.1 多重纹理概述 30919.1.1 启用多重纹理
31119.1.2 多重纹理坐标 31219.2 Pixel Shader的输入和输出 31319.3 使用Pixel Shader的步骤 31419.3.1 编写
并编译Pixel Shader 程序 31419.3.2 创建一个Pixel Shader 31419.3.3 设置一个Pixel Shader 31519.3.4 销毁一
个Pixel Shader 31519.4 HLSL取样对象 31519.5 示例：通过Pixel Shader实现 多重纹理 31719.6 小结 324第20
章 特效框架 (Effects Framework) 32520.1 Technique与Pass 32620.2 更多的HLSL内建对象 32720.2.1 texture对
象 32720.2.2 取样器(sampler)对象和取样器状态 32820.2.3 Vertex Shader对象和Pixel Shader对象 32820.2.4
字符串(string) 32920.2.5 注解(Annotations) 33020.3 特效文件中的设备状态 33020.4 创建一个特效 33120.5
设置常量 33320.6 使用特效 33520.6.1 获得特效句柄 33520.6.2 激活特效 33620.6.3 启用特效 33620.6.4 设置
当前的渲染Pass 33720.6.5 关闭特效 33720.6.6 示例 33720.7 示例：通过特效文件实现光照效果和纹理贴图
33820.8 示例：雾化效果 34320.9 示例：卡通效果 34520.10 特效编辑器(EffectEdit) 34720.11 小结 347附录A
Windows程序设计简介 349A.1 概述 349A.1.1 资源 350A.1.2 事件、消息队列、消息和消息循环 350A.1.3
图形用户接口(GUI) 351A.2 简单的Windows应用程序 352A.3 一个更好的消息循环 364A.4 小结 365参考
文献 366

章节摘录

插图：在现实中的许多表面都类似于镜面，可以在这些表面上看到物体的影像。在本节中将讲述如何在三维场景中模拟镜面效果。为了减少问题的复杂性，这里只实现简单的平面镜面效果。例如，一辆擦亮的轿车也可以反射出物体的影像，但轿车的车身是平滑的弧形表面，而不是一个简单的平面。这里模拟的反射现象类似于擦亮的大理石地板或者是挂在墙上的镜子，它们也可以反射物体的影像，一句话，本节只讨论平面的镜面效果。在程序中实现镜子的功能需要解决两个问题。第一，必须知道如何求出对于一个任意表面的对称物体，然后才可以正确地画出反射的物体。第二，只能在镜子中显示反射的物体影像。也就是说，必须把某个表面标记为镜子，表明如果反射的物体影像位于镜子平面中才会对它进行渲染。第一个问题通过一些向量的几何知识来解决，使用模板缓冲区可以解决第二个问题。以下两个小节中将分别解释如何解决这两个问题，第三个小节将两个问题综合在一起，然后讨论本章的第一个简单例子“镜面效果”的相关代码。

编辑推荐

本教材是网络游戏编程的专业课程。《DirectX游戏编程》以游戏开发中的图形技术为主线，着重介绍如何使用DirectX9.0开发交互式3D图形程序。通过对DirectX的基本概念和使用的详细描述，帮助学习者掌握游戏开发界高级三维图形技术，为成为优秀游戏图形工程师打好基础。第九艺术学院丛书作为RGDC游戏开发课程体系配套教材，由递归教育组织力量，集国内外游戏业内精英人才打造而成。全套丛书共包括：游戏开发基础、游戏美术设计、游戏程序开发3个部分。游戏开发核心技术深度剖析计算机图形API游戏开发真体验中国文化创意产业技术创新联盟中国电影电视技术学会数字特效与三维动画专业委员会中国系统仿真学会数字娱乐专业委员会推荐教材

精彩短评

- 1、质量很好没有发现缺点
- 2、把龙书从头到尾抄一遍

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:www.tushu000.com