中国电子教育学会高教分会推荐
普通高等教育新工科数字媒体专业“十三五”课改规划教材

三维编程原理及Direct3D实践

编 著
宋 伟 刘子澍 田逸非

西安电子科技大学出版社
http://www.xduph.com

中国电子教育学会高教分会推荐

普通高等教育新工科数字媒体专业“十三五”课改规划教材

三维编程原理及Direct3D实践

宋　伟　刘子澍　田逸非　编著

西安电子科技大学出版社

内 容 简 介

本书主要讲解 DirectX 9.0 的三维编程知识，包括 DirectX 简介、基本空间变换、Direct3D 的绘制方法、Alpha 融合、光照与材质、三维网格模型、拾取、动画网格模型、使用 DirectX 绘制文字、自由摄像机、Sprite、粒子系统、音效播放以及基于 TCP/IP 的网络游戏基础等内容。

本书可以作为高等学校数字媒体技术专业游戏开发方向相关必修课的教材，也可以作为本科计算机专业相关选修课的教材，还可以作为对计算机图形学感兴趣或者希望了解游戏引擎底层原理的读者的参考书籍。

图书在版编目(CIP)数据

三维编程原理及 Direct3D 实践 / 宋伟，刘子澍，田逸非编著. —西安：
西安电子科技大学出版社，2019.8
ISBN 978-7-5606-5409-6

Ⅰ. ①三… Ⅱ. ①宋… ②刘… ③田… Ⅲ. ①DirectX 软件—程序设计—高等学校—教材
Ⅳ. ①TP317

中国版本图书馆 CIP 数据核字(2019)第 157448 号

策划编辑　刘小莉
责任编辑　马晓娟
出版发行　西安电子科技大学出版社(西安市太白南路 2 号)
电　　话　(029)88242885　88201467　　邮　　编　710071
网　　址　www.xduph.com　　电子邮箱　xdupfxb001@163.com
经　　销　新华书店
印刷单位　陕西日报社
版　　次　2019 年 8 月第 1 版　2019 年 8 月第 1 次印刷
开　　本　787 毫米×1092 毫米　1/16　印 张 11.375
字　　数　241 千字
印　　数　1～2000 册
定　　价　26.00 元

ISBN 978 - 7 - 5606 - 5409 - 6 / TP

XDUP 5711001-1

作者简介

宋伟，北方工业大学信息学院副教授，韩国东国大学多媒体工学专业博士生导师。

2005年毕业于东北大学软件工程专业，获学士学位；2013年毕业于韩国东国大学多媒体工学系，获工学博士学位。2013年至今，工作于北方工业大学信息学院，现任国际学院副院长。近年的研究领域主要涉及激光雷达、三维重建、并行计算、无人驾驶、虚拟现实等。发表学术论文100余篇，其中SCI检索期刊论文27篇；发明专利3项；出版专著1部。主持横、纵向项目10余项，包括国家自然科学基金1项、教育部留学回国人员科研启动基金1项、北京市留学人员科技活动择优资助1项。获北京市高等教育教学成果二等奖1项。

前　言

DirectX 集成了 Direct3D、DirectDraw、DirectInput、DirectPlay、DirectSound、DirectShow、DirectSetup、DirectMediaObjects 等多个组件。Direct3D 是基于微软的通用对象模式 COM 的 3D 图形 API。DirectX SDK 当前常见的版本有 DirectX 9、DirectX 10、DirectX 11、DirectX 12。DirectX 9 具有良好的硬件兼容性，可以支持当前绝大部分主流显卡，具有复杂的顶点着色引擎。DirectX 10 在几何处理阶段增加了几何渲染单元等功能，提升了 GPU 效率，可以提供更精细的模型细节。DirectX 11 和 DirectX 12 在之前版本的基础上，新增了计算着色器等功能，图像质量比 DirectX 9 高很多。

作为基础版本，DirectX 9 所涉及的三维编程基本原理，如三维顶点缓存的创建和使用、三维空间的变换、光照和材质等在后续版本中未发生变化。DirectX 9 中自带的数学库 D3DX 在后续版本中被“抛弃”，取而代之的是一个专门为向量计算进行过优化的 XNA 库。新版本中大量使用着色器，老版本中一些已有的光照等效果在新版本中使用时需要编写着色器文件。对于 3ds Max 模型的程序载入，DirectX 9 提供了一个加载以.x 结尾的三维模型文件的函数接口，可以直接利用它加载从 3ds Max 等建模软件里导出的.x 模型文件，用于三维物体或者骨骼动画的学习和使用。为提高底层三维编程技能，本书使用 DirectX 系列中十分经典的 DirectX 9 进行讲解。

本书的内容编排如下：

第 1 章　DirectX 简介，介绍 Direct3D 开发环境配置和三维场景绘制的实现过程，并通过案例讲述基于面向对象思想的 Direct3D 开发过程模块封装方法。

第 2 章　基本空间变换，介绍计算机图形学的空间变换原理和算法。

第 3 章　Direct3D 的绘制方法，介绍利用三维顶点、颜色、纹理等元素创建三维模型并绘制的方法。

第 4 章　Alpha 融合，介绍利用 Alpha 通道的透明渲染方法。

第 5 章　光照与材质，介绍 Direct3D 的光照原理和物体材质的创建及使用方法。

第 6 章　三维网格模型，介绍从 3ds Max 的模型导出 XFile 文件的方法以及将其载入

Direct3D 环境的方法和渲染的程序实现过程，并讲解通过三维模型的边界检测原理实现物体碰撞的过程。

第 7 章　拾取，介绍利用射线与物体相交判断鼠标是否点击到场景中的三维物体的方法。

第 8 章　动画网格模型，介绍蒙皮动画的原理，讲解如何将骨骼动画数据从 3ds Max 中导出为 XFile 文件，并实现其在 Direct3D 环境中的加载和渲染。

第 9 章　使用 DirectX 绘制文字，主要介绍在场景中使用 ID3DXFont 接口绘制二维文字以及使用 ID3DXMesh 接口绘制三维文字的方法。

第 10 章　自由摄像机，介绍第一人称自由摄像机的原理和实现方法，以及如何使用鼠标和键盘对摄像机进行控制。

第 11 章　Sprite，介绍使用 ID3DXSprite 接口实现 Sprite 的绘制与移动的方法。

第 12 章　粒子系统，介绍使用 Sprite 实现二维粒子系统的方法以及利用点 Sprite 实现粒子枪的方法。

第 13 章　音效播放，介绍通过 DirectSound 加载并播放 WAV 格式的音频文件的方法。

第 14 章　基于 TCP/IP 的网络游戏基础，简单介绍了 TCP/IP 通信协议，主要讲解在 Windows 平台下使用 Socket 实现不同客户端通过服务器进行通信的方法。

本书详细介绍了三维编程过程中常用的 DirectX 接口，并为读者提供了基本的参数使用规则，若对本书内容有任何疑问，欢迎致信 tianyifei0000@sina.com。本书中所涉及的代码均为开源。

除了 3 位作者之外，参与本书编纂工作的还有北方工业大学信息学院张凌峰、邱吕扬、韩金昆、孙溯、寥金巧、高焱宁、王世麟等同学。

本书的出版得到了北方工业大学 2018 年教育教学改革和课程建设研究项目、北方工业大学“毓优”人才项目、国家自然科学基金项目(61503005)、北京市“长城学者”培养计划项目(CIT&TCD20190304)等资助。

注：书中代码可在出版社网站下载。

北方工业大学　宋伟

2019 年 6 月

目　录　CONTENTS

第一部分　三维编程基础

第二部分 三维编程应用

第一部分

三维编程基础

第 1 章　DirectX 简介

DirectX 是由微软开发的一套 API，这套 API 包含了 DirectGraphics(Direct3D+DirectDraw)、DirectInput、DirectShow、DirectSound、DirectPlay、DirectSetup、DirectMediaObjects 等多个组件，它可使程序员在编程时不用关心电脑底层硬件。本书重点介绍 DirectX 的图形 API——Direct3D。

1.1　Direct3D 程序启动

Direct3D，简称 D3D，是一套底层图形 API，被视作应用程序与图形设备交互的中介。借助该 API，我们能够利用其硬件加速功能来绘制 3D 场景。

用 DirectX 开发程序无需关心所使用的硬件设备，用 Direct3D 开发游戏也可以不用关心电脑使用的图形硬件，这主要得益于图形硬件设备之上的硬件抽象层(Hardware Abstraction Layer，HAL)，它极大地方便了游戏开发中的程序编写。HAL 是由硬件制造商提供的设备驱动程序接口，Direct3D 可以通过 HAL 直接与图形硬件通信。此外，利用 HAL 提供的图形硬件加速功能，Direct3D 可以绘制出更高效和更高质量的游戏场景。另外，在 DirectX 和 HAL 之间还存在一个硬件模拟层(Hardware Emulation Layer，HEL)。当电脑上的硬件不支持 Direct3D 的某些高级功能时，HEL 会通过软件运算来模拟硬件运算(后面将介绍 Direct3D 初始化时如何获取 Direct3D 设备；检测硬件设备是否支持我们提出的一些性能要求；在要求不能满足时，系统将采用软件进行运算模拟支持，以达到和硬件支持同样的效果)。

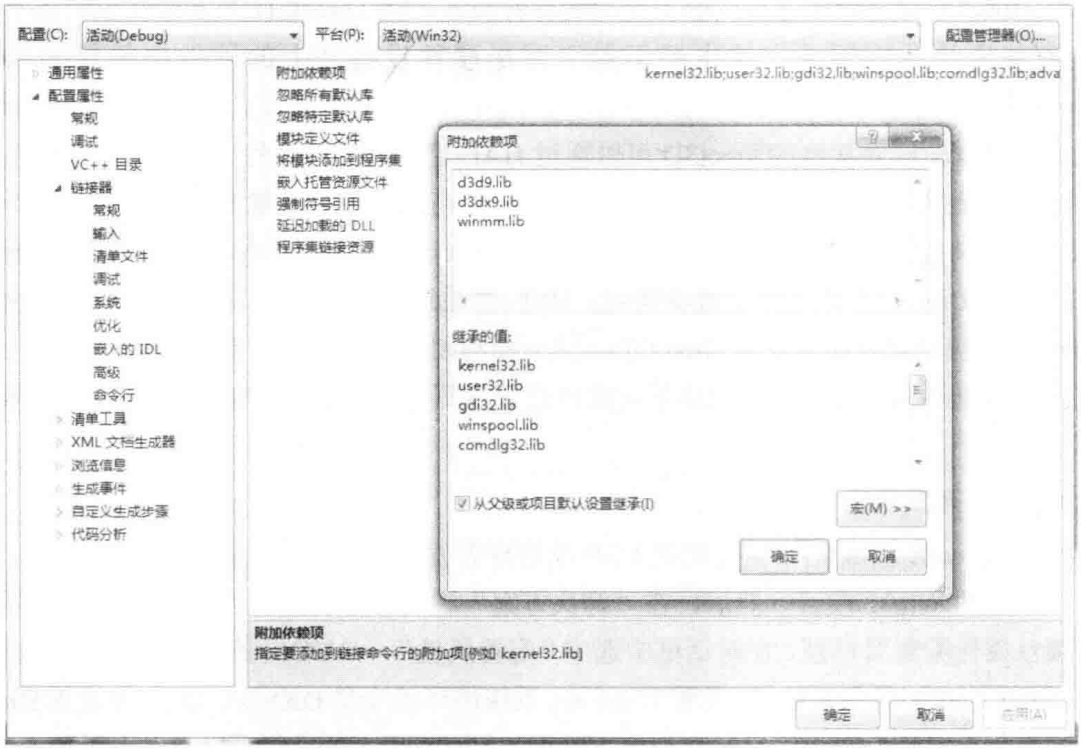
1．配置开发环境

安装 DirectX SDK 后，需要配置 D3D 的程序开发环境。首先，打开 Visual Studio 2010 创建一个空的 Win32 项目。然后，在“解决方案”栏选中项目名，单击右键的属性菜单，弹出项目属性配置对话框。在对话框中选中“配置属性”→“VC++目录”，如图 1.1(a)所示。根据 DirectX SDK 的安装路径，设置 Windows 系统的环境变量 DXSDK_DIR，并设置 DirectX 项目的“包含目录”和“库目录”位置。如图 1.1(b)所示，在“包含目录”中添加 DirectX 头文件目录“$(DXSDK_DIR)Include”。查看工程所用编译器，若工程所选编译器为 Win32，则在“库目录”中添加 DirectX 库文件目录“$(DXSDK_DIR)Lib\x86”；若工程所选编译器

为 x64，则在“库目录”中添加“$(DXSDK_DIR)Lib\x64”。最后，在链接器下的“输入”选项中选择“附加依赖项”，添加 d3d9.lib、d3dx9.lib、winmm.lib 文件，如图 1.1(c)所示。至此完成了 DirectX 的 Windows 开发环境配置。

(a) 项目属性配置对话框

(b) “包含目录”和“库目录”的 DirectX SDK 路径设置



(c) Direct3D Lib 文件设置

图 1.1　Direct3D 开发环境的配置

2. 初始化

在程序开始，需要对 Direct3D 进行初始化，其步骤如图 1.2 所示。

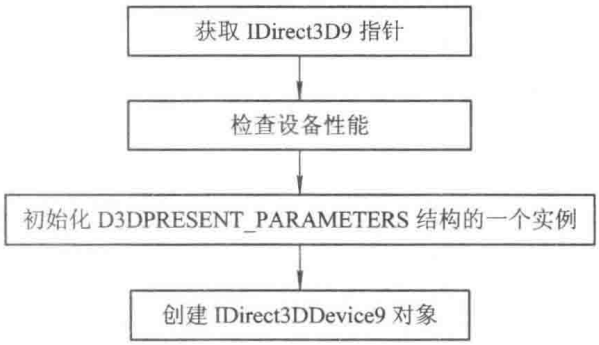


图 1.2　Direct3D 的初始化过程

1) 获取 IDirect3D9 指针

IDirect3D9 接口指针用于设备枚举和创建 IDirect3DDevice9 对象。设备枚举可以获取当前系统中每块可用的物理硬件设备的信息。创建 IDirect3DDevice9 对象是指根据我们需要的功能和获取的设备性能，创建一种用来显示 3D 图形的硬件设备对象。以下代码片段展示了如何创建 IDirect3D9 接口对象的指针：

```
IDirect3D9*   d3d9 = 0;
d3d9 = Direct3DCreate9(D3D_SDK_VERSION);
```

在 Direct3D 编程中，IDirect3D9 是我们需要创建的第一个对象，也是我们最后释放的对象。需要注意的是，Direct3DCreate9 的参数是 D3D_SDK_VERSION，它指定了我们创建的 IDirect3D9 指针使用的 SDK 版本。

2) 检查设备性能

为检查设备性能及获取硬件设备信息，需利用 GetDeviceCaps()函数及其返回值，将设备信息保存到一个 D3DCAPS9 结构中。如果显卡不支持硬件顶点运算，需要利用 HEL 层(在 Direct3D 和 HAL 层之间)执行软件顶点运算来模拟硬件顶点运算。虽然软件顶点运算比硬件顶点运算慢，但是它可以保证我们的程序不会因为不支持某功能而中断。在创建 IDirect3DDevice9 之前，必须明确显卡是否支持硬件顶点运算功能。利用以下代码可以判断主显卡是否支持硬件顶点运算功能：

```
D3DCAPS9 caps;
d3d9->GetDeviceCaps(D3DADAPTER_DEFAULT, DeviceType, &caps);
int vp = 0;
if( caps.DevCaps & D3DDEVCAPS_HWTRANSFORMANDLIGHT )
    vp = D3DCREATE_HARDWARE_VERTEXPROCESSING;
else
    vp = D3DCREATE_SOFTWARE_VERTEXPROCESSING;
```

3) 初始化 D3DPRESENT_PARAMETERS 结构的一个实例

D3DPRESENT_PARAMETERS 结构由诸多属性组成，可以通过指定变量类型来创建 IDirect3DDevice9 的接口。该结构的原型如下：

```
typedef struct _D3DPRESENT_PARAMETERS_
{
    UINT                BackBufferWidth;
    UINT                BackBufferHeight;
    D3DFORMAT           BackBufferFormat;
    UINT                BackBufferCount;

    D3DMULTISAMPLE_TYPE MultiSampleType;
    DWORD               MultiSampleQuality;

    D3DSWAPEFFECT       SwapEffect;
    HWND                hDeviceWindow;
    BOOL                Windowed;
    BOOL                EnableAutoDepthStencil;
    D3DFORMAT           AutoDepthStencilFormat;
    DWORD               Flags;

    UINT                FullScreen_RefreshRateInHz;
    UINT                PresentationInterval;
} D3DPRESENT_PARAMETERS;
```

各项参数的含义如下：

- BackBufferWidth：后台缓存区表面的宽度，单位为像素。
- BackBufferHeight：后台缓存区表面的高度，单位为像素。
- BackBufferFormat：后台缓存像素格式，如 D3DFMT_R8G8B8、D3DFMT_A8R8G8B8 等。
- BackBufferCount：后台缓存区的个数。
- MultiSampleType：后台缓存使用的多重采样类型。
- MultiSampleQuality：多重采样的质量水平。
- SwapEffect：指定交换链中缓存的页面置换方式，其值为 D3DSWAPEFFECT 枚举类型中的一个成员，可指定为 D3DSWAPEFFECT_DISCARD、D3DSWAPEFFECT_FLIP 或 D3DSWAPEFFECT_COPY 常量。

• hDeviceWindow：与设备相关的窗口句柄，指定了要进行绘制的应用程序窗口。

• Windowed：窗口显示模式，为 true 时代表窗口模式，为 false 时代表全屏模式。

• EnableAutoDepthStencil：为 true 时，Direct3D 自动创建并维护深度缓存或模板缓存。

• AutoDepthStencilFormat：深度缓存或模板缓存的像素格式。在后面的示例中我们将其设为 D3DFMT_D24S8，用 24 位表示深度，8 位保留给模板缓存用。

• Flags：一些附加的属性，可以指定为 0 (无标记)或 D3DPRESENTFLAG 集合中的一个成员。

• FullScreen_RefreshRateInHz：刷新频率，设置为 D3DPRESENT_RATE_DEFAULT 时表示使用默认的刷新频率。

• PresentationInterval：D3DPRESENT 集合的一个成员。设置为 D3DPRESENT_INTERVAL_IMMEDIATE 时表示立即显示更新；设置为 D3DPRESENT_INTERVAL_DEFAULT 时，表示由 Direct3D 来选择提交频率。通常该值等于刷新频率。

下面这段代码是一个 D3DPRESENT_PARAMETERS 结构体的初始化实例：

```
D3DPRESENT_PARAMETERS d3dpp;
d3dpp.BackBufferWidth = width;
d3dpp.BackBufferHeight = height;
d3dpp.BackBufferFormat = D3DFMT_A8R8G8B8;
d3dpp.BackBufferCount = 1;
d3dpp.MultiSampleType = D3DMULTISAMPLE_NONE;
d3dpp.MultiSampleQuality = 0;
d3dpp.SwapEffect = D3DSWAPEFFECT_DISCARD;
d3dpp.hDeviceWindow = hwnd;
d3dpp.Windowed = windowed;
d3dpp.EnableAutoDepthStencil = true;
d3dpp.AutoDepthStencilFormat = D3DFMT_D24S8;
d3dpp.Flags = 0;
d3dpp.FullScreen_RefreshRateInHz = D3DPRESENT_RATE_DEFAULT;
d3dpp.PresentationInterval = D3DPRESENT_INTERVAL_IMMEDIATE;
```

4) 创建 IDirect3DDevice9 对象

我们利用上面设定好的 D3DPRESENT_PARAMETERS 结构体实例，通过 CreateDevice() 函数创建 IDirect3DDevice9 接口对象。CreateDevice()函数的原型如下：

```
HRESULT CreateDevice(
  UINT                    Adapter,
  D3DDEVTYPE              DeviceType,
```