

图书基本信息

书名：<<DirectX 10 3D游戏编程深度探索>>

13位ISBN编号：9787302249207

10位ISBN编号：7302249202

出版时间：2011-9

出版时间：清华大学出版社

作者：沃尔什

页数：513

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：<http://www.tushu007.com>

内容概要

Peter Walsh的这本《DirectX 10

3D游戏编程深度探索》内容涵盖面广，讲解深入浅出，且示例丰富。

书中主要介绍如何使用DirectX 开发交互式

3D图形程序，重点是游戏开发。

全书首先介绍了必要的入门知识，如开发平台、图形学基础、数学工具，然后讲解了相关的3D概念。

其他主题几乎涵盖了Direct3D中的所有基本运算，例如图元的绘制、光照、纹理、alpha融合、模板，以及如何使用Direct3D实现游戏所需的技术。

《DirectX 10

3D游戏编程深度探索》可供从事3D游戏程序设计、可视化系统设计或其他图形应用程序开发的开发人员和大中专院校学生参考，也极适合各种游戏开发培训机构用作Direct3D编程的培训教程。

作者简介

Peter

Walsh是一位资深软件工程师，目前供职于苏格兰邓迪的Realtime Worlds公司，他参与了即将推出的游戏A.P.B.的开发。

不久前，他刚参与了为Xbox

360开发的Crackdown的游戏引擎的研制。

此外。

他参与过的游戏项目还包括Medal of Honor : Rising

Sun、Harry Potter and the Chamber of Secrets.FI Career

Challenge以及Time of Defiance。

他还是5本游戏开发书籍的作者，这些书中就包括本书的上一版Advanced 3D Game Programming with DirectX 9, 0(《DirectX 9.0 3D游戏高级编程》)。

书籍目录

第1章 关于Windows

- 1.1 关于Windows
- 1.2 匈牙利命名法
- 1.3 Windows的一般概念
- 1.4 Windows中的消息处理
- 1.5 Hello World——Windows风格
- 1.6 对窗口几何参数的操作
- 1.7 重要的窗口消息
- 1.8 类的封装
- 1.9 COM：组件对象模型
- 1.10 小结

第2章 DirectX 10入门

- 2.1 什么是DirectX
- 2.2 安装
- 2.3 VC++的环境设置
- 2.4 DirectDraw发展史
- 2.5 Direct3D简介
 - 2.5.1 2D图形学基础
 - 2.5.2 纹理
 - 2.5.3 用cGraphicsLayer启动Direct3D
 - 2.5.4 Direct3D的初始化
 - 2.5.5 关闭Direct3D
- 2.6 例程：Direct3D Sample
- 2.7 小结

第3章 输入与音效

第4章 3D数学基础

第5章 人工智能

第6章 基于UDP的多玩家Internet网络游戏

第7章 Direct3D基础

第8章 高级3D技术

第9章 Direct3D高级主题

第10章 场景管理

附录 STL入门

章节摘录

版权页：插图：应用程序实例（另外一个句柄，代表一个运行的实例）。

每个进程都有不同的实例句柄，用来唯一标识Windows的进程。

这与窗口句柄是不同的，每个应用程序可能控制多个窗口。

而您却需要牢牢抓住这个实例，因为Windows API函数需要知道是哪个实例在调用它。

我们可以把实例看成可执行的程序在内存中的一个副本或映像。

每个可执行程序都有一个句柄，这样Windows才能够区分、管理或对它们进行其他操作。

HINSTANCE类型，当前运行的应用程序的另一个副本。

在以往机器的内存很小，Windows中的一个程序可能有多个实例来共享一段内存空间。

现在每个进程都有自己独立的内存空间，那么这个参数则总是设置为NULL。

该参数之所以还保留下来，是为了让以往的Windows应用程序能够继续工作。

指向命令行字符串的指针。

当用户把一个文件拖到资源管理器中的一个可执行文件上时（而非该程序的一个运行副本）

，Windows便会运行该程序，并把被拖动文件的路径和文件名作为命令行的第一个参数。

一组标志，用于描述窗口应如何被初始化（如全屏、最小化等）。

从概念上来说，WinMain函数的工作流程如下：WinMain注册Windows的应用程序类 创建主窗口 while（没有通知退出）处理Windows发送的所有消息 MyRegisterClass（）接收应用程序的实例，并向Windows通知该应用程序（其实就是注册）。

InitInstallce（）用于创建屏幕上的主窗口，并开始绘制。

然后，代码将进入while循环，直到程序退出。

GetMessage（）函数用来从消息队列中取得消息。

该函数总是返回1，除非它在队列中遇到如下这个特定消息：“现在退出”消息，其消息ID是WM_QUIT。

只要消息队列中还有消息存在，GetMessage（）就会从队列中移出消息，并用它来填充消息结构体，即前面提到的msg变量。

在while循环内部，您首先需要做的是获取消息，并使用TranslateMessage（）函数翻译该消息。

这是一个非常便捷的函数。

当接收到某个按键被按下或者释放的消息时，您将得到该按键对应的虚拟键码。

这些虚拟键码的实际ID值可以是任意的，但是其命名空间是我们所关心的：当字母a被按下的时候，其中相应的消息参数就等价于#define VK_A。

当您想要做一些像处理文本输入一类的事情时，上述的命名法处理起来就会使人非常痛苦。

TranslateMessage（）就可以给您帮忙了，它可以把参数从VK_A转化为(char)'a'，这就简化了对常规文本输入的处理。

对于那些没有等价ASCII码的键，如Page UP和LeftArrow，在TranslateMessage（）函数中将继续保留它们的虚拟键码（分别为VK_PRIOR和VK_LEFT）。

其他所有的消息经过TranslateMessage（）处理后，都不会有任何改变。

第二个函数DispatchMessage（），是一个实际处理消息的函数。

在内部，它会查看哪个函数（在MyRegisterClass中）被注册用于处理消息，然后再把消息传递给那个函数。

您将注意到，我们的程序从来没有真正调用过窗口过程。

这是因为当您把它指定给DispatchMessage（）数时，Windows已经为您调用了。

编辑推荐

《DirectX 10 3D游戏编程深度探索》可供从事3D游戏程序设计、可视化系统设计或其他图形应用程序开发的开发人员和大中专院校学生参考，也极适合各种游戏开发培训机构用作Direct3D编程的培训教程。

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:<http://www.tushu007.com>