

SAMS



OpenGL

超级宝典 (第三版)

〔美〕Richard S. Wright, Jr. Benjamin Lipchak 著
徐波 译



人民邮电出版社
POSTS & TELECOM PRESS

OpenGL 超级宝典 (第三版)

[美] Richard S. Wright, Jr. Benjamin Lipchak 著

徐 波 译

人民邮电出版社

图书在版编目 (CIP) 数据

OpenGL 超级宝典: 第3版 / (美) 赖特 (Wright, R.S.), (美) 利普查克 (Lipchak, B.) 著; 徐波译. —北京: 人民邮电出版社, 2005.9

ISBN 7-115-13720-X

I. 0... II. ①赖...②利...③徐... III. 图形软件, OpenGL IV. TP391.41

中国版本图书馆 CIP 数据核字 (2005) 第 101125 号

版 权 声 明

Richard S. Wright, Jr. Benjamin Lipchak: OpenGL SuperBible, Third Edition (ISBN: 0-672-32601-9)

Copyright ©2005 by Sams Publishing

Authorized translation from the English language edition published by the Sams Publishing

All rights reserved.

本书中文简体字版由美国 Sams 出版公司授权人民邮电出版社出版。未经出版者书面许可, 对本书任何部分不得以任何方式复制或抄袭。

版权所有, 侵权必究。

OpenGL 超级宝典 (第三版)

- ◆ 著 [美] Richard S. Wright, Jr. Benjamin Lipchak
译 徐 波
责任编辑 陈冀康
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京鸿佳印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16
印张: 47.25
字数: 1 469 千字 2005 年 9 月第 1 版
印数: 1-4 000 册 2005 年 9 月北京第 1 次印刷

著作权合同登记号 图字: 01-2004-4008 号

ISBN 7-115-13720-X/TP · 4840

定价: 88.00 元 (附光盘)

读者服务热线: (010) 67132692 印装质量热线: (010) 67129223

本书是 OpenGL 编程指南,涵盖了使用新版本的 OpenGL 进行编程所需要的主要知识。

全书分 3 个部分,共 23 章,另有 3 个附录。第一部分包括第 1 章到第 12 章,介绍经典 OpenGL 绘图的所有基础知识,包括 3D 图形和 OpenGL 简介,OpenGL 空间绘图,几何转换,颜色、材料和光照,纹理贴图,曲线和表面,管线,交互式图形等内容。第二部分包括第 13 章到第 15 章,分别介绍了不同操作系统平台上的 OpenGL 功能特性和编程细节。第三部分是第 16 章到第 23 章,着重介绍 OpenGL 的一些高级功能以及 OpenGL 1.5 的新特性。附录部分给出了更多阅读建议、术语表和对 OpenGL ES 的简单介绍。随书光盘中包含了不同操作系统平台下的丰富的示例程序及其源代码。

本书适合于希望精通 OpenGL 以便对他们的图形编程和 3D 图形知识进行扩展的程序员,也可以帮助那些经验丰富的 OpenGL 程序员学习如何移植自己的应用程序。本书既可以作为学习 OpenGL 的教材,也可以作为随时查阅的参考手册。

Richard S. Wright, Jr.

当 OpenGL 首次在 Windows 平台上亮相时，Richard 就开始接触 OpenGL，至今已有近十年的历史。目前，他在佛罗里达州的奥兰多 Full Sail 学院的游戏设计系从事 OpenGL 编程教学，并且是 Starstone Software Systems 公司的总裁。这家公司使用 OpenGL 为 PC 和 Macintosh 平台开发多媒体模拟软件。

Richard 曾就职于 Real 3D/Lockheed Martin，常常参加 OpenGL ARB 会议，并对 OpenGL 1.2 规范以及一致性测试做出了自己的贡献。从那时开始，Richard 就致力于多维数据库可视化、游戏开发、医学诊断可视化以及天文空间模拟等方面的研究。

Richard 从 1978 年开始学习编程，当时他还是一位八年级的学生，并且是在纸终端上学习编程。16 岁那年，父母没有给他买汽车，而是给他买了一台电脑。不到一年的时间，他就卖出了自己的第一个计算机程序（这是个图形程序！）。高中毕业之后，他的第一份工作是在当地的一家消费教育公司从事编程和计算机文化的教学。他曾在路易斯威尔大学的 Speed Scientific 学校学习电子工程和计算机科学。高年级的时候，他半途离校去了佛罗里达，并迎来了他职业生涯的高峰。他原籍是肯塔基州的路易斯威尔，现在和妻子以及他们的三个孩子一起住在奥兰多和代纳托比奇（Daytona Beach）之间。在不编程或躲避飓风的时候，他是位热心的业余天文爱好者。此外，他还在一家星期日学校担任教师。

Benjamin Lipchak

Benjamin 毕业于温彻斯特理工学院（Worcester Polytechnic Institute），主修技术写作和计算机科学。“为什么拥有了计算机科学学位还需要成为作家？”这个问题是当他接受 DEC 的一个技术写作职位面试时被问及的。对于 Benjamin 而言，这是一个意义重大的上午。他的面试时间比预定的要长。在结束面试时，他得到的邀请是加入 DEC 的 Alpha 工作站 OpenGL 驱动程序开发队伍。

在 Compaq 收购了 DEC 并遣散了绝大多数图形开发小组之后，Benjamin 在一个星期五去职，但随即又在星期一归来，接受了一个条件更优越的工作合同。尽管地位有变，Benjamin 在 Compaq 继续担任领导角色，经常走访顾客，并且管理与 PowerStorm UNIX 和 NT OpenGL 驱动程序有关的项目。虽然工作的主要焦点是工作站应用程序，但他的团队在游戏领域也有很大突破，在 AlphaStations 上运行的 GLQuake 达到了前所未有的速度。

Internet 的蓬勃发展使 Benjamin 受到了诱惑，他曾经脱离本行两年，创立了一家和图像搜索有关的公司。Internet 泡沫的破灭使他又回到了最初所

爱的 OpenGL, 并就职于位于马萨诸塞州马尔波罗的 ATI Research。最近, 他在片断着色器领域极为活跃, 这增加了对 Radeon OpenGL 驱动程序的支持。Benjamin 在负责形成 GL_ARB_fragment_program 扩展规范的 OpenGL ARB 工作小组中担任负责人, 并参与了 ATI 的 OpenGL 着色语言的开发。在非常有限的空余时间里, 他总会设法外出参加徒步旅行或划船。他还创立了一家独立的唱片公司 Wachusett Records, 主营钢琴独奏乐曲。

徐波, 浙江宁波人, 熟悉 C、C++ 和 Java 语言。2002 年开始从事计算机技术图书翻译, 第一部译作《C 专家编程》博得好评, 其谦逊平和的态度也使读者产生好感。译作《C 和指针》也深受广大读者的喜爱。徐波技术视野广阔, 翻译文笔优美, 是颇具潜力的译作者。

我第一次听说 OpenGL 是在 1992 年的 San Francisco Win32 开发人员会议上，当时 Windows NT 3.1 还处于早期的 Beta 版本（或晚期的 Alpha）阶段。许多厂商参加了会议，并纷纷表态将对这个激动人心的图形新技术提供支持。在众多厂商之中，有一家叫做 SGI (Silicon Graphics, Inc) 的公司。SGI 公司的代表展示了他们的图形工作站，并播放了一些特效视频演示，这些特效都来自于一些流行的电影。但是，他们到会的主要目的是推广一种新的叫做 OpenGL 的 3D 图形标准。它基于 SGI 专用的 IRIS GL，并进行了革新，使之成为一种图形标准。引人注目的是，Microsoft 也表示要在 Windows NT 上对 OpenGL 提供支持。

直到 Windows NT 3.5 的 Beta 版本发布，我才第一次亲自体验 OpenGL。最初那些基于 OpenGL 的屏幕保护程序揭开了这种图形 API 的序幕。和许多人一样，除了努力在 Microsoft 的帮助文件中搜索之外，我还购买了一本《OpenGL 编程指南》¹（现被许多人简称为“红皮书”）。但是，这本红皮书并不是入门读物，它假定读者已经具备了大量基本知识（当时的我对这些知识恰好一无所知）。

现在，我必须坦白一件事，就是我是如何学习 OpenGL 的。我是通过编写一本这方面的书来学习它的。是的，它就是《OpenGL 超级宝典》的第一版。我是在出版商的最后期限的压力之下，通过编写这本书来学习 3D 图形的。但是，我还是努力完成了这个任务。1996 年，本书的第一版面世了。从头开始教自己学习 OpenGL 使我能够采用一种更好的方式来解释 OpenGL 的 API，这种方式也是许多人所喜欢的。在本书的出版过程中，Waite Group Press 被另一家出版社所收购，从而导致整个项目差点夭折。Mitchell Waite 仍然坚持原来的立场，认为 OpenGL 将成为计算机图形的“下一个巨人”。他的信念收到了回报，因为本书很快就要求紧急重印，第一次印刷的书几乎没有运到仓库就已售完。

这已经是很久以前的事情了，看上去是那么的遥远……

仅仅 3 年以后，3D 加速的图形即使在最普通的 PC 上也已经成为主流。Microsoft 和 SGI 之间的一场“API 战争”来了又去。OpenGL 坚实地奠定了它在 PC 世界的地位，3D 硬件加速也已经像 CD-ROM 和声卡那么普通。当我在 Lockheed Martin/Real 3D 工作时，我设法使自己的职业方向朝着 OpenGL 靠近，并为 OpenGL 规范的 1.2 版本作出了自己的一点贡献。本书的第二版在 1999 年末出版，它对第一版进行了显著的扩展和修正。我们甚至做了一些尝试，通过使用 GLUT 框架，使本书的所有示例程序在非 Windows 平台上变得更为友好。

¹ 编者注：本书中文版《OpenGL 编程指南（第四版）》已由人民邮电出版社于 2005 年 4 月出版。

现在,差不多又过了5年(离本书的第一版已有8年),我们推出了本书的第三版。现在,OpenGL毫无疑问已经是一种占据主导地位的实时3D图形API。今天,即使在最为简陋的PC上,OpenGL仍然能够体现出它超强的稳定性和卓越的性能。OpenGL也是UNIX和Linux操作系统的图形标准,Apple更是在新的MacOS X操作系统中使OpenGL成为一种核心的基本技术。OpenGL甚至通过一种新的规范OpenGL ES向嵌入式和移动领域发动攻势。在5年之前,谁会想到Quake游戏能够在手机上运行呢?

今天,令人激动的是即使是笔记本电脑也拥有了3D加速功能,OpenGL真正深入到每种主流计算平台。但是,更加激动人心的是计算机图形硬件的持续发展。今天,绝大多数图形硬件都是可编程的,OpenGL甚至推出了它自己的着色语言(shading language),可以产生令人惊异的逼真图形。在20世纪的硬件平台上,这几乎是难以想像的(在很多时候,我不得不屈服于硬件上的限制)。

在第三版中,我很高兴Benjamin Lipchak能够成为我的合作者。Benjamin主要负责和OpenGL着色器程序有关的章节。他来自ARB中专门负责OpenGL这方面内容的小组,是这个领域最具有发言权的人之一。

我们还完全去除了第一版“特定于Microsoft”的特征,而是采取了一种多平台的方式。本书所有的编程例子都在Windows、MacOS X以及至少一个版本的Linux上进行了测试。我们还为每种操作系统分别设立了一章,介绍在这些平台的本地应用程序中使用OpenGL的相关信息。

本书内容

《OpenGL 超级宝典(第三版)》分为三个部分。首先,我们讨论所谓的经典OpenGL。这是OpenGL从一开始就具有的固定管线功能。很多章节在第二版的基础上进行了非常大的扩展,因为OpenGL在1.2版本之后进行了一些修订。在很长一段时间内,固定管线编程仍然会和我们在一起,它的简单编程模型使它仍然是许多有多年开发经验的程序员的选择。

在这些章节中,你将学习用OpenGL进行实时3D图形编程的一些基础知识。你将学习如何创建一个使用OpenGL的程序,如何设置3D渲染环境,以及如何创建基本物体,并对它们进行光照和着色。然后,我们将进一步深入,对OpenGL的一些高级特性和不同特效进行更深入的探讨。这些章节很好地向你介绍了如何用OpenGL进行3D图形编程,并提供了一些概念性的基础知识,为本书后面章节所介绍的一些更为高级的功能打下了基础。

第二部分的3章分别介绍了在3种主流的操作平台(Windows、MacOS X和Linux/UNIX)上使用OpenGL的一些特定信息。最后,第三部分所包含的内容不仅是OpenGL的最新特性,也是今天3D图形硬件的最新特性。尤其是OpenGL的着色语言,它是OpenGL 2.0的主要特性,展现了计算机图形多年以来的最大发展。

第一部分 经典 OpenGL

第1章——3D图形和OpenGL简介

这个介绍性的章节专门面对3D图形的新手。它介绍了一些基本概念和常用词汇。

第2章——使用OpenGL

在本章中,我们向你介绍了什么是OpenGL、它的来源以及它的未来。你将学习用OpenGL编写第一个程序,找到你需要使用的头文件和函数库,学习如何设置环境以及发现一些能够帮助记忆OpenGL函数调用的常见约定。我们还介绍了OpenGL的状态机和错误处理机制。

第3章——在空间中绘图(几何图元和缓冲区)

在这一章中,我们讨论了3D图形编程的一些基本成分。你将学习如何命令一台计算机用OpenGL创建一个三维物体。你还将学习隐藏表面消除的基本知识,并学习使用模版缓冲区的方法。

第 4 章——几何转换——管线

既然已经在虚拟空间中创建一些三维形状，该如何移动它们呢？该如何移动自己的观察位置呢？这些就是你将在本章学习的内容。

第 5 章——颜色、材料和光照基础知识

在本章中，将对你所创建的三维“轮廓”进行处理，为它们提供颜色。你将学习如何在图形上应用材料效果和光照，使它们看上去更为真实。

第 6 章——颜色和材料的更多细节

在本章中，你将学习如何把物体与背影进行混合，以创建透明的物体（能够看穿）。还将学习使用雾和累积缓冲区所实现的一些特殊效果。

第 7 章——OpenGL 中的图像

本章介绍如何在 OpenGL 中操纵图像数据。这些信息包括读取一个 TGA 文件，并在一个 OpenGL 窗口中显示它。还将学习一些强大的 OpenGL 图像处理功能。

第 8 章——纹理贴图基础知识

纹理贴图是所有 3D 图形工具箱最为有用的特性之一。你将学习如何把图像包装到多边形上，并学习如何同时载入和管理多重纹理。

第 9 章——纹理贴图高级知识

在本章中，你将学习如何自动生成纹理坐标，使用高级过滤模式以及使用内置的对纹理压缩的硬件支持。还将学习 OpenGL 纹理组合器的强大功能。

第 10 章——曲线和表面

简单的三角形是一种功能强大的基本成份。本章介绍一些工具，对万能的三角形进行操纵。你将学习 OpenGL 的一些内置的二次方程表面生成函数，并且学习如何使用自动镶嵌功能把复杂的形状分解为更小、更容易处理的片段。还将学习一些工具函数，产生 Bézier、NURBS 曲线和表面，可以使用这些函数，用非常少量的代码创建复杂的形状。

第 11 章——这就是管线——更快的几何图形渲染

在本章中，我们向你展示如何用一些更小、更简单的 3D 物体创建复杂的 3D 物体，介绍 OpenGL 的显示列表和顶点数组，它们可以提高性能，并对你的模型进行有效的组织。还将学习如何进行详细的分析，显示如何用最好的方式表示大型的、复杂的模型。

第 12 章——交互性图形

本章解释了两种 OpenGL 特性：选择和反馈。和这两种特性有关的函数允许用户与场景中的物体进行交互。你还可以获得和场景中单个物体有关的渲染细节。

第二部分 OpenGL 无处不在

第 13 章——Windows 上的 OpenGL——Wiggle

在本章中，你将学习编写真正的使用 OpenGL 的 Windows（消息驱动的）程序。你将学习 Windows 的“Wiggle”函数，把 OpenGL 的渲染代码粘贴到 Windows 设备环境上。还将学习如何对 Windows 的消息做出响应。

第 14 章——MacOS X 上的 OpenGL

在本章中，你将学习如何在本地的 MacOS X 应用程序中使用 OpenGL。本章的示例程序向你展示了如何使用 Xcode 开发环境在 Carbon 或 Cocoa 下开始工作。

第 15 章——Linux 上的 OpenGL

本章讨论 GLX，也就是用于在 UNIX 和 Linux 的 X Window 系统上支持 OpenGL 应用程序的 OpenGL 扩展。你将学习如何创建和管理 OpenGL 渲染环境，以及如何用几个常用的 GUI 工具箱创建 OpenGL 绘图区域。

第三部分 下一代 OpenGL

第 16 章——缓冲区对象

在本章中,你将学习 OpenGL 1.5 的顶点缓冲区对象特性。缓冲区对象允许你把顶点数组数据存储在可以由 GPU 进行高效访问的内存中,例如本地 VRAM 或 AGP 映射的系统内存中。

第 17 章——遮挡查询

在本章中,你将学习 OpenGL 1.5 的遮挡查询机制。这个特性允许你有效地对场景中的物体执行代价很小的测试渲染,判断它们是否将被其他物体所遮挡。如果情况属实,就可以不绘制它们,从而节省时间。

第 18 章——深度纹理和阴影

本章讨论 OpenGL 1.4 的深度纹理和阴影比较。你将学习如何在场景中引入实时的阴影效果,而与几何图形的复杂度无关。

第 19 章——可编程管线

抛弃旧的,迎来新的。在介绍新的可编程顶点和片断管线阶段之前,本章首先回顾了常规的固定功能管线。可编程管线允许你对渲染进行定制。在着色器程序出现之前,这个任务是无法实现的。

第 20 章——低层着色

在本章中,你将学习低层的着色器扩展: ARB_vertex_program 和 ARB_fragment_program。你可以使用它们对你的渲染任务进行定制,在着色器程序中使用一种类似于汇编语言的代码,它可以提供对底层硬件的完全控制。

第 21 章——高层着色

在本章中,我们讨论 OpenGL 的着色语言,其功能等价于低层扩展。GLSL 是一种类似 C 的语言,它向你提供了增强的功能和效率。

第 22 章——顶点着色

本章通过一些简单的例子描述了顶点着色器的使用,包括光照、雾、压缩和扩张,以及换皮的功能。

第 23 章——片断着色

本章也是通过例子来学习的。这次使用的是各种片断着色器。这些例子包括以像素为基础的光照、颜色转换、图像处理 and 过程性纹理。有些例子还使用了顶点着色器。这些例子在现实世界中具有代表性的用途,你可以从中找到顶点着色器和片断着色器协同工作的场合。

关于本书所附的 CD

《OpenGL 超级宝典 (第三版)》所附的 CD 包括一些示例程序、工具箱、源代码以及相关文档。我们不断把自己发掘出来的东西放到 CD 上,一直到本书的出版。请阅读 CD 根目录下的 readme.txt 文件,查看这张 CD 所包括内容的完整清单。

- \Examples——在这个目录下,你将会发现本书的每章都有一个子目录,包含了该章的示例程序。每个程序都有一个真正的名称(而不是像 sample 5.2c 这样的名称),因此你可以轻松地浏览 CD,运行认为有趣的任何程序。

- \Tools——这个目录包含了一些第三方的工具以及函数库。每个都有它自己的子目录以及来自原厂商的文档。本章的大部分示例程序都使用了其中一些工具(尤其是 GLUT)。

- \Demos——这个目录包含了一些 OpenGL Demo 程序。这些程序都用于显示 OpenGL 的渲染功能。有些程序是免费的,有些是商业软件的 Demo 版本。

有关本书 CD 的支持问题,请与 Sams Publishing (www.sampublishing.com) 联系。当然,你还可以

从本书的网站 <http://www.starstonesoftware.com/OpenGL> 找到其他 OpenGL 问题、更多的示例程序和教程，以及难以避免的勘误表。

创建示例程序

本书所有的示例程序都是用 C 编写的。当然我们可以很轻松地把它转换为 C++ 程序，但对于一本像这样的书籍，用 C 编写示例程序能够拥有更广泛的读者群，并且也很容易被那些更乐于使用 C++ 的人所理解。绝大多数程序还使用了 glTools 函数库，它是作者所编写的一些友好的 OpenGL 工具函数的集合。glTools 头文件 (gltools.h) 和 .c 源文件位于 CD 的 \Examples 目录下的 \common 目录。

在 \Examples 目录下，你将会找到每种操作系统平台所支持的目录。Windows 项目文件是用 Visual C++ 6.0 编写的。原因是很多人不愿意把他们的开发环境升级到新的基于 .Net 的版本，而那些已经使用新版本的人（包括我，已经用它编写了许多项目）也可以很轻松地导入这些项目。

MacOS X 示例程序的项目文件是用 Xcode 1.1 版本创建的。那些仍然使用 Project Builder 的人应该升级了。Xcode 是免费的，而且可以使你的生活变得更轻松。所有的 Mac 示例程序都在 OS X 版本 10.3.3 上进行了测试。在本书付印的时候，Apple 还没有发布支持 OpenGL 着色语言的驱动程序。当它推出这个驱动程序的时候，一些潜在的 bug 可能会表现出来。请根据前面所提到的网站，了解任何必要的更新。

本书还提供了 Linux 示例程序的 make file。配置一种 Linux 环境的方法可以多达 10392444224229349244281999.4 种。也许真正管用的没那么多，但毫不夸张地说，你完全可以按照自己的想法来做。由于 Xcode 使用了一种类似的 gnu 编译器（它和许多 Linux 环境所使用的编译器相同），因此你不会遇到太多问题。我希望本书的网站不久就能提供更多的注意事项和相关的教程。

在 3D 中大胆地想像

在 20 世纪 80 年代早期的一天，我在一家电子商店想寻找一台计算机。一位售货员走近我，向我推销他的产品。我告诉他我刚刚开始学习编程，想购买一台 Amiga，而不是他所提供的型号。他告诫我应该认真对待全世界的人都在使用的计算机。他告诉我，Amiga 除了“产生一些漂亮的图片”之外就别无用处。他使我相信，没有人可以依靠计算机所产生的漂亮图片来谋生。

这也许是我所听从的所有糟糕建议中最糟糕的一个。几年前，我忘记了自己曾是一位“令人尊敬的”数据库/企业级……开发人员。现在，我编写非常酷的图形程序，教授图形编程，拥有自己的软件公司，觉得自己所从事的是法律所允许的工作中最能找到乐趣的！

现在我希望能够向你提供一些更好的建议。不论你想编写游戏、创建军事模拟程序、开发科学可视化应用或实现大型公司数据库的可视界面，OpenGL 都是一种完美的 API。当你还是一位新手的时候，它可以满足你的需要；当你已经是一位 3D 高手时，它仍然可以增强你的力量。是的，朋友，你可以依靠计算机所生成的图片舒舒服服地过日子！

——Richard S. Wright, Jr.

第一部分 经典 OpenGL

第 1 章 3D 图形和 OpenGL 简介	3
1.1 计算机图形的简单历史回顾.....	3
1.1.1 进入 CRT.....	3
1.1.2 走向 3D.....	4
1.2 3D 效果初探.....	6
1.2.1 透视 (视角).....	6
1.2.2 颜色和着色.....	6
1.2.3 光照和阴影.....	7
1.2.4 纹理贴图.....	7
1.2.5 雾.....	7
1.2.6 混和和透明.....	8
1.2.7 抗锯齿.....	8
1.3 3D 图形的常见用途.....	9
1.3.1 实时 3D.....	9
1.3.2 非实时 3D.....	10
1.4 3D 编程的基本原则.....	11
1.4.1 立即模式和保持模式 (场景图).....	11
1.4.2 坐标系.....	11
1.4.3 投影 (从 3D 到 2D).....	14
1.5 小结.....	16
第 2 章 使用 OpenGL	17
2.1 什么是 OpenGL.....	17
2.1.1 标准的演化.....	18
2.1.2 API 战争.....	19
2.2 OpenGL 的工作原理.....	22
2.2.1 泛型实现.....	22
2.2.2 硬件实现.....	23
2.2.3 管线.....	24
2.3 OpenGL 是 API 而不是编程语言.....	24
2.4 API 规范.....	25
2.4.1 数据类型.....	25
2.4.2 函数名约定.....	26

2.5 平台独立性.....	27	3.9.4 多边形创建规则.....	79
2.5.1 使用 GLUT.....	27	3.9.5 细分和边界.....	80
2.5.2 设置编程环境.....	28	3.10 其他缓冲区技巧.....	82
2.5.3 你的第一个程序.....	28	3.10.1 使用缓冲区目标.....	82
2.5.4 用 OpenGL 绘制形状.....	32	3.10.2 操纵深度缓冲区.....	83
2.6 使用 OpenGL 和 GLUT 生成动画.....	37	3.10.3 用剪刀进行裁剪.....	84
2.7 OpenGL 状态机.....	40	3.10.4 使用模版缓冲区.....	85
2.8 OpenGL 错误.....	41	3.11 小结.....	88
2.9 确认版本.....	42	3.12 参考.....	88
2.10 调用 glHint 获得线索.....	42	第 4 章 几何转换——管线.....	100
2.11 使用扩展.....	42	4.1 本章是不是枯燥的数学课.....	100
2.11.1 检查扩展.....	43	4.2 理解转换.....	101
2.11.2 这是谁的扩展.....	43	4.2.1 视觉坐标.....	101
2.11.3 在 Windows 中使用 OpenGL 1.1 以后的版本.....	44	4.2.2 视图转换.....	102
2.12 小结.....	44	4.2.3 模型转换.....	102
2.13 参考.....	45	4.2.4 模型视图二元性.....	103
第 3 章 在空间中绘图 (几何图元和 缓冲区).....	55	4.2.5 投影转换.....	104
3.1 在 3D 中绘制点.....	55	4.2.6 视口转换.....	104
3.2 设置 3D 画布.....	56	4.3 矩阵——3D 图形的数学基础.....	105
3.3 3D 空间中的点——顶点.....	57	4.3.1 什么是矩阵.....	105
3.4 绘制图形.....	58	4.3.2 转换管线.....	105
3.5 设置点的大小.....	60	4.3.3 模型视图矩阵.....	106
3.6 在 3D 空间中画直线.....	62	4.3.4 单位矩阵.....	108
3.6.1 线带和线环.....	63	4.3.5 矩阵堆栈.....	109
3.6.2 用直线构成近似曲线.....	64	4.3.6 一个原子例子.....	110
3.6.3 设置直线的宽度.....	64	4.4 使用投影.....	112
3.6.4 直线点画.....	66	4.4.1 正投影.....	112
3.7 在 3D 空间中绘制三角形.....	68	4.4.2 透视投影.....	113
3.7.1 三角形——第一种多边形.....	68	4.4.3 一个远处缩小的 例子.....	114
3.7.2 环绕.....	68	4.5 高级矩阵操作.....	116
3.7.3 三角形带.....	69	4.5.1 载入矩阵.....	117
3.7.4 三角形扇.....	69	4.5.2 自己执行转换.....	118
3.8 创建实心物体.....	70	4.5.3 把转换放在一起.....	120
3.8.1 设置多边形颜色.....	72	4.6 使用照相机和角色在 OpenGL 中移动.....	121
3.8.2 隐藏表面消除.....	72	4.6.1 角色帧.....	121
3.8.3 剔除: 隐藏表面, 提高性能.....	73	4.6.2 欧拉角: “卢克! 请使用帧”.....	122
3.8.4 多边形模式.....	75	4.6.3 照相机管理.....	122
3.9 其他图元.....	75	4.7 综合应用.....	123
3.9.1 四条边的多边形——四边形.....	75	4.8 小结.....	127
3.9.2 通用多边形.....	76	4.9 参考.....	128
3.9.3 填充多边形, 回顾点画.....	76		

第 5 章 颜色、材料和光照基础知识	134	5.9.5 法线均衡	157
5.1 什么是颜色	134	5.10 综合讨论	158
5.1.1 光的波性	134	5.10.1 创建亮点	158
5.1.2 光的粒子性	135	5.10.2 绘制光点	159
5.1.3 人眼作为光子检测器	136	5.11 阴影	162
5.1.4 计算机作为光子检测器	136	5.11.1 什么是阴影	163
5.2 PC 颜色硬件	137	5.11.2 实现压平的代码	163
5.3 PC 显示模式	137	5.11.3 一个阴影例子	164
5.3.1 屏幕分辨率	138	5.11.4 重新回顾球体世界	166
5.3.2 颜色深度	138	5.12 小结	167
5.4 在 OpenGL 中使用颜色	139	5.13 参考	167
5.4.1 颜色立方体	139	第 6 章 颜色和材料的更多细节	174
5.4.2 设置绘图颜色	140	6.1 混合	174
5.4.3 着色	140	6.1.1 组合颜色	174
5.4.4 设置着色模型	142	6.1.2 修改混合方程式	177
5.5 现实世界的颜色	142	6.1.3 抗锯齿	177
5.5.1 环境光	143	6.2 雾	181
5.5.2 散射光	143	6.3 累积缓冲区	183
5.5.3 镜面光	143	6.4 其他颜色操作	185
5.5.4 综合考虑	143	6.4.1 颜色掩码	185
5.6 现实世界的材料	144	6.4.2 颜色逻辑操作	185
5.6.1 材料属性	144	6.4.3 Alpha 测试	186
5.6.2 向材料添加光照	144	6.4.4 抖动	186
5.6.3 计算环境光效果	144	6.5 小结	186
5.6.4 散射和镜面光效果	145	6.6 参考	187
5.7 向场景添加光照	145	第 7 章 OpenGL 中的图像	191
5.7.1 启用光照	145	7.1 位图	191
5.7.2 设置光照模型	146	7.2 像素包装	195
5.7.3 设置材料属性	146	7.3 像素图	196
5.8 使用光源	148	7.3.1 经过包装的像素格式	198
5.8.1 哪种方式最合适	148	7.3.2 一个颜色更丰富的例子	198
5.8.2 表面法线	149	7.3.3 移动像素	200
5.8.3 指定法线	149	7.3.4 保存像素	201
5.8.4 单位法线	150	7.4 像素的更多乐趣	203
5.8.5 寻找法线	151	7.4.1 像素缩放	208
5.8.6 设置光源	152	7.4.2 像素转移	208
5.8.7 设置材料属性	153	7.4.3 像素映射	211
5.8.8 指定多边形	153	7.5 图像“子集”	212
5.9 光照效果	154	7.6 小结	222
5.9.1 镜面加亮	154	7.7 参考	223
5.9.2 镜面光	154	第 8 章 纹理贴图基础知识	239
5.9.3 镜面反射	155	8.1 载入纹理	239
5.9.4 镜面指数	156		

8.1.1 使用颜色缓冲区	242	10.3.4 NURBS 属性	323
8.1.2 更新纹理	242	10.3.5 定义表面	323
8.2 将纹理贴图到几何图形	243	10.3.6 修剪	324
8.3 一个简单的 2D 例子	244	10.3.7 NURBS 表面	326
8.4 纹理环境	249	10.4 镶嵌	326
8.5 纹理参数	250	10.4.1 镶嵌器	327
8.5.1 基本过滤	250	10.4.2 镶嵌器回调函数	327
8.5.2 纹理环绕	251	10.4.3 指定顶点数据	328
8.5.3 带纹理的卡通	252	10.4.4 综合应用	328
8.5.4 Mip 贴图	256	10.5 小结	332
8.6 纹理对象	259	10.6 参考	332
8.7 小结	266		
8.8 参考	266		
第 9 章 纹理贴图高级知识	279		
9.1 第二颜色	279	第 11 章 这就是管线—更快的几何 图形渲染	353
9.2 各向异性过滤	281	11.1 模型装配 101	353
9.3 纹理压缩	282	11.2 显示列表	363
9.3.1 压缩纹理	283	11.2.1 批处理	363
9.3.2 载入压缩纹理	284	11.2.2 预批处理	364
9.4 纹理坐标生成	284	11.2.3 使用显示列表的一些 注意事项	365
9.4.1 物体线性贴图	289	11.2.4 转换到显示列表	365
9.4.2 视觉线性贴图	290	11.3 测量性能	366
9.4.3 球体贴图	291	11.4 顶点数组	370
9.4.4 立方体贴图	292	11.4.1 载入几何图形	373
9.5 多重纹理	294	11.4.2 启用数组	374
9.5.1 多重纹理坐标	295	11.4.3 数据在哪里	374
9.5.2 一个多重纹理例子	295	11.4.4 进行绘图	375
9.6 纹理组合器	298	11.4.5 索引顶点数组	376
9.7 小结	300	11.5 小结	385
9.8 参考	300	11.6 参考	385
第 10 章 曲线和表面	306	第 12 章 交互式图形	395
10.1 内部支持的表面	306	12.1 选择	395
10.1.1 设置二次方程状态	307	12.1.1 为图元命名	396
10.1.2 绘制二次方程图形	308	12.1.2 在选择模式下工作	397
10.1.3 用二次方程进行建模	311	12.1.3 选择缓冲区	398
10.2 Bézier 曲线和表面	312	12.1.4 挑选	399
10.2.1 参数方程表示形式	313	12.1.5 层次式挑选	401
10.2.2 求值器	314	12.1.6 反馈	404
10.3 NURBS	321	12.1.7 反馈缓冲区	404
10.3.1 从 Bézier 到 B 样条	321	12.1.8 反馈数据	405
10.3.2 结点	322	12.1.9 用户定义标记	405
10.3.3 创建 NURBS 表面	322	12.2 一个反馈例子	406
		12.2.1 对物体加上标签以便反馈	406

12.2.2	步骤 1: 选择物体	408		
12.2.3	步骤 2: 从物体获取 反馈信息	409		
12.3	小结	411		
12.4	参考	411		
第二部分 OpenGL 无处不在				
第 13 章 Windows 上的 OpenGL Wiggle 417				
13.1	Windows 上的 OpenGL 实现	417		
13.1.1	泛型 OpenGL	418		
13.1.2	可安装客户驱动程序 (ICD)	418		
13.1.3	Mini 客户驱动程序 (MCD)	418		
13.1.4	Mini 驱动程序	419		
13.1.5	扩展 OpenGL	419		
13.2	基本的窗口渲染	420		
13.2.1	GDI 设备环境	420		
13.2.2	像素格式	421		
13.2.3	OpenGL 渲染环境	426		
13.3	综合应用	427		
13.3.1	创建窗口	427		
13.3.2	使用 OpenGL 渲染环境	430		
13.3.3	其他 Windows 消息	433		
13.4	Windows 调色板	434		
13.4.1	颜色匹配	434		
13.4.2	调色板仲裁	435		
13.4.3	为 OpenGL 创建调色板	436		
13.4.4	调色板的创建和销毁	439		
13.5	OpenGL 和 Windows 字体	440		
13.5.1	3D 字体和文本	441		
13.5.2	2D 字体和文本	442		
13.6	全屏渲染	444		
13.6.1	创建一个无边框窗口	444		
13.6.2	创建一个全屏窗口	445		
13.7	多线程渲染	447		
13.8	OpenGL 和 WGL 扩展	448		
13.8.1	简单扩展	448		
13.8.2	使用新入口	449		
13.8.3	WGL 扩展	449		
13.9	小结	470		
13.10	参考	470		
第 14 章 MacOS X 上的 OpenGL 482				
14.1	基础知识	482		
14.1.1	框架	482		
14.1.2	使用 GLUT API	483		
14.2	使用 AGL 和 Carbon API	483		
14.2.1	像素格式	483		
14.2.2	管理渲染环境	484		
14.2.3	执行双缓冲渲染	485		
14.2.4	你的第一个 AGL 程序	485		
14.2.5	使用位图字体	494		
14.3	使用 Cocoa API	504		
14.3.1	NSOpenGL 类	504		
14.3.2	你的第一个 Cocoa 程序	505		
14.4	小结	512		
14.5	参考	512		
第 15 章 GLX: Linux 上的 OpenGL 516				
15.1	基础知识	516		
15.1.1	使用 OpenGL 和 X11 函数库	516		
15.1.2	使用 GLUT 函数库	518		
15.1.3	Linux 上的 OpenGL	518		
15.1.4	OpenGL 模拟: Mesa	519		
15.2	X Window 系统的 OpenGL 扩展	519		
15.2.1	X Window 系统基础知识	519		
15.2.2	选择画面	519		
15.2.3	管理 OpenGL 渲染环境	520		
15.2.4	创建 OpenGL 窗口	521		
15.2.5	双缓冲窗口	521		
15.2.6	综合应用	521		
15.3	为 OpenGL 创建位图字体	528		
15.4	离屏渲染	537		
15.4.1	使用 GLX Pixmap	537		
15.4.2	使用 Pbuffer	541		
15.5	使用 Motif 函数库	546		
15.5.1	GLwDrawingArea 和 GLwMDrawingArea	546		
15.5.2	回调资源	547		
15.5.3	函数	548		
15.5.4	综合应用	549		
15.6	小结	557		
15.7	参考	557		

第三部分 下一代 OpenGL

第 16 章 缓冲区对象	565	19.1.1 固定的顶点处理	609
16.1 顶点数组.....	566	19.1.2 固定的片断处理	611
16.1.1 生成球状微粒云彩	566	19.2 新的占主导	612
16.1.2 启用顶点数组	567	19.2.1 可编程顶点着色器	612
16.1.3 使用更多的球体	568	19.2.2 固定功能胶水	614
16.2 迁移到缓冲区对象	569	19.2.3 可编程片断着色器	614
16.2.1 缓冲区对象管理	570	19.3 着色器扩展介绍	615
16.2.2 用缓冲区对象进行渲染	571	19.3.1 低层扩展	615
16.3 把数据载入到缓冲区对象	571	19.3.2 高层扩展	617
16.3.1 把数据复制到缓冲区对象	571	19.4 小结	618
16.3.2 直接映射缓冲区对象	572	第 20 章 低层着色 (深入底层)	619
16.4 补遗	575	20.1 管理低层着色器	619
16.5 小结	576	20.1.1 创建和绑定着色器	619
16.6 参考	576	20.1.2 载入着色器	620
第 17 章 遮挡查询	581	20.1.3 删除着色器	621
17.1 遮挡查询之前的世界	581	20.1.4 设置扩展	621
17.2 边框	584	20.2 指令集	622
17.3 对查询对象进行查询	587	20.2.1 常用指令	622
17.4 小结	589	20.2.2 专用于顶点的指令	623
17.5 参考	589	20.2.3 专用于片断的指令	624
第 18 章 深度纹理和阴影	593	20.3 变量类型	624
18.1 作为光源	594	20.3.1 临时变量	625
18.1.1 使场景正好占据整个窗口	594	20.3.2 参数	625
18.1.2 去掉华而不实的东西	594	20.3.3 属性	626
18.2 一种新类型的纹理	597	20.3.4 输出	628
18.3 首先绘制阴影	597	20.3.5 别名	629
18.4 然后是光照	598	20.3.6 地址	629
18.4.1 投影阴影贴图:		20.4 输入和输出限定符	630
“为什么”	598	20.4.1 输入取负	630
18.4.2 投影阴影贴图:		20.4.2 输入 Swizzle	630
“怎么做”	599	20.4.3 输出写入掩码	630
18.4.3 阴影比较	600	20.4.4 输出限定	630
18.5 只用两个就够了	605	20.5 资源消耗和队列	631
18.6 关于多边形偏移	606	20.5.1 解析器限制	631
18.7 小结	606	20.5.2 本地限制	632
18.8 参考	607	20.5.3 其他查询	633
第 19 章 可编程管线	609	20.6 着色器选项	633
19.1 旧的被淘汰	609	20.6.1 位置不变性顶点选项	634
		20.6.2 雾应用片段选项	634
		20.6.3 精度提示片断选项	634
		20.7 小结	634
		20.8 参考	634