PEARSON
Addison Wesley

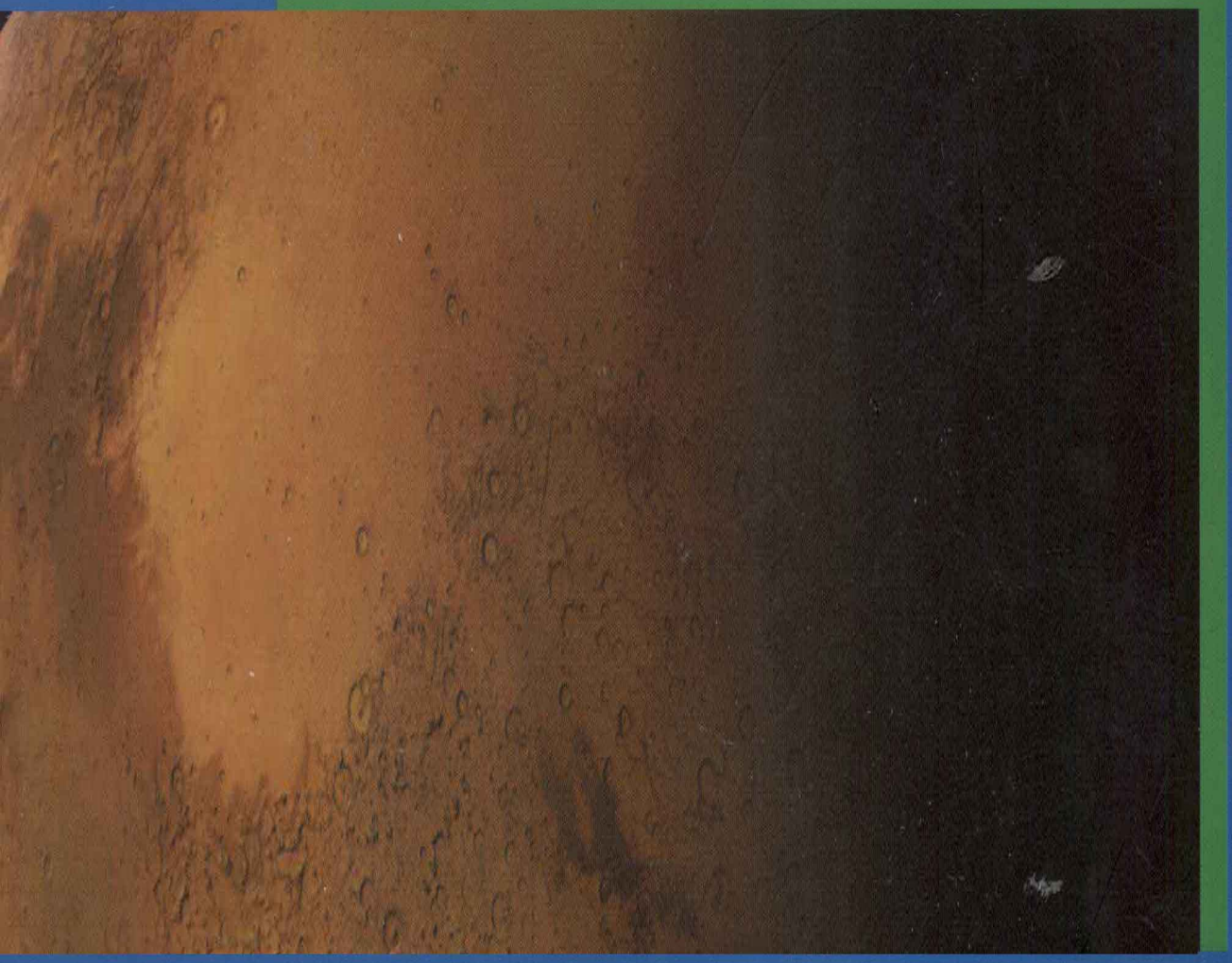
OpenGL 超级宝典（第5版）

丰富 权威 实用

OpenGL® SuperBible Fifth Edition

〔美〕Richard S. Wright, Jr. Nicholas Haemel
Graham Sellers Benjamin Lipchak 著

付飞 李艳辉 译



人民邮电出版社
POSTS & TELECOM PRESS

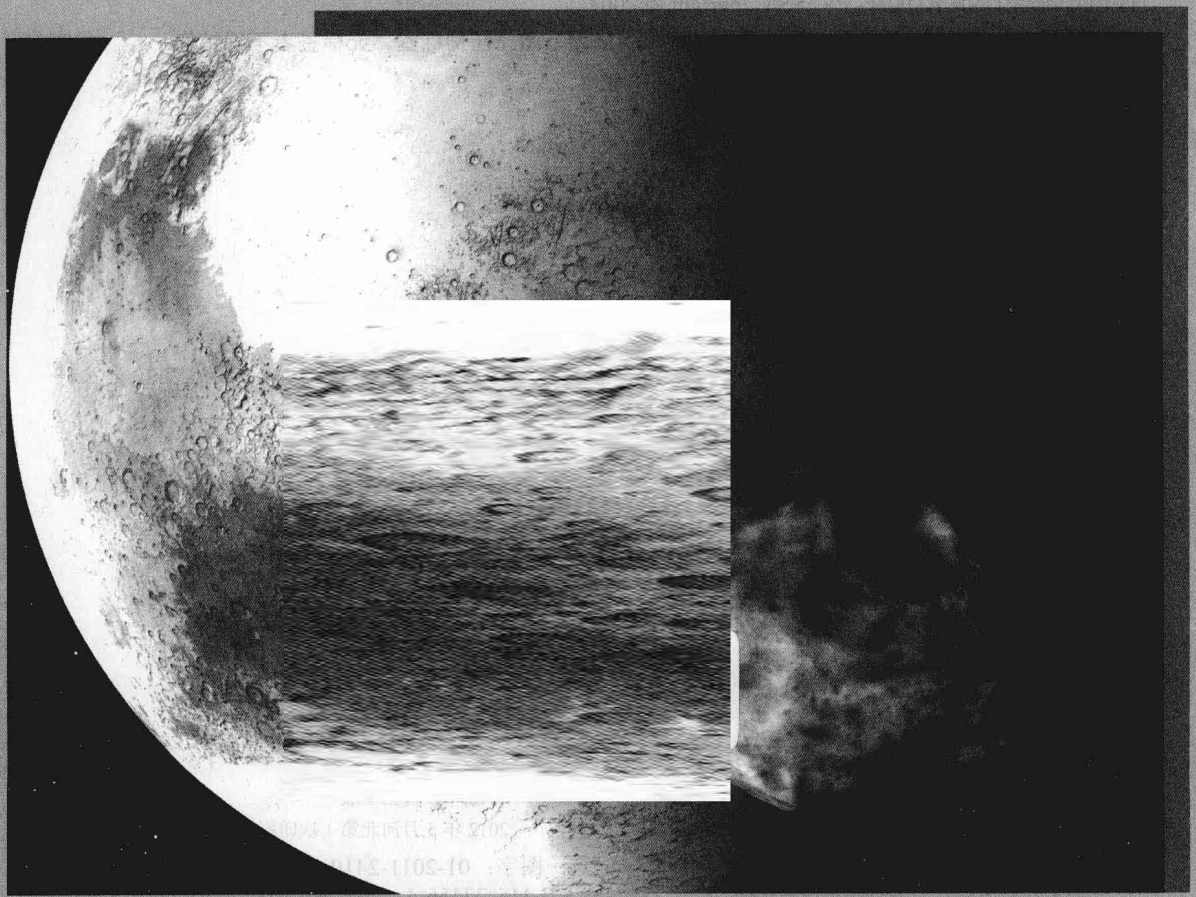
PEARSON
Addison Wesley

OpenGL 超级宝典（第5版）

OpenGL® SuperBible Fifth Edition

〔美〕Richard S. Wright, Jr. Nicholas Haemel
Graham Sellers Benjamin Lipchak 著

付飞 李艳辉 译



人民邮电出版社

北京

内容提要

本书是 OpenGL 及 3D 图形编程最好的入门指南，涵盖了使用最新版本的 OpenGL 进行编程所需要的主要知识。

全书分三部分，共 16 章，另有 3 个附录。第一部分包括第 1 章到第 7 章，介绍如何构建一个使用 OpenGL 的程序、如何设置 3D 渲染环境，以及如何创建基本对象和光线并对他们进行着色。然后深入研究如何使用 OpenGL，并向读者介绍 GLSL，以及如何创建自己的着色器。第二部分包括第 8 章到第 12 章，将进行更深入的研究，而懂得如何应用这些高级特性将使读者超越业余 3D 玩家的水平。这一部分不仅能够使我们掌握更多的可视化效果，同时也考虑了性能表现。第三部分包括第 13 章到第 16 章，着重介绍 OpenGL 如何支持和连接 Windows、Mac OS X、Linux 和掌上设备。附录部分给出了更多阅读建议、术语表和 API 参考介绍。

本书适合希望精通 OpenGL 以便对图形编程和 3D 图形知识进行扩展的程序员阅读，也可以帮助经验丰富的 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。

序

在自然界中，有时候一片森林会长得过于茂密，以至于它的生态系统不堪承受自身的负担而濒临崩溃，这片森林可能会被闪电击中，这样往往会出现一个崭新的开始，并在以前的基础上展开一个新的蓝图。本书的第 5 版就经历了一次类似的根本性转变，我们已经彻底抛弃了第 4 版，只保留了一小部分独立内容，使这本书呈现了新的生机。

对于 OpenGL 来说也是如此，在 OpenGL 3.0 中，首次将“不推荐”（deprecated）这个词引入到规范中，对很多特性和功能都进行了删除标记，并强烈鼓励开发者转向新的技术，预期得到的是一个更加简洁清晰的 OpenGL 版本，可以看成是从开发包中抛弃了固定管线部分，然而，事情并不总是按照预期的情况发展。在上世纪 90 年代后期，OpenGL 陷入了一场现在所谓的“API 战争”（API Wars），这是由微软公司的 Direct 3D API 引起的，但是，开发者最终会选择更加优越的 API，而 OpenGL 拒绝消亡；相反，它变得更加繁荣，并且成为了实时 3D 图形渲染的世界性标准。开发者似乎又一次表明了态度，无论 ARB 中 OpenGL 的捍卫者如何努力，看来固定管线都拒绝消亡。现在，我们可以选择两种 OpenGL，即兼容版本（compatibility profile）和核心版本（core profile）。

出于这点考虑，由于本书的第 4 版覆盖了现在可以称为“经典 OpenGL 2.1 固定管线版本”的内容，在未来的几年内完全有理由继续受欢迎，固定功能管线拥有大量的传统优势。它的编程非常容易，并且目前能够在现代图形加速卡上进行完全的硬件加速，在未来一段时期内，很多性能要求不高和非图形专业的专家可能会非常推荐这种编程模型，只有时间能证明一切。

同时，我们还面临着在本书的第 5 版中该怎样做的问题，每隔 6 个月左右，OpenGL 就会进行一次升级，而在本书编写过程中我们就经历了两次升级，如果试图同时覆盖兼容版本和核心版本的内容，则会产生很多混淆之处，另外，很多更新、更现代的效果只能在着色器上实现。随着时间一年一年地推移，很多人还在使用的图形编程的固定管线模式看来是越来越过时了。因此，考虑到技术的进步，我们认为在第 5 版中最好专注于介绍核心版本，就我所知，这将是第一本这样做的图书。当我在佛塞大学（或译为全航大学，Full Sail University）讲授 OpenGL 时，我在试图寻找如何讲授不包含固定管线的 OpenGL 时遇到了巨大的挑战。我和所注意到的其他很多人都是将着色器编程作为固定管线的一个扩展来讲授的，我们怎样才能跳过在刚开始的时候非常麻烦的关于着色器编程的几章内容，而先开始使用 OpenGL 呢？有很多工具可以帮助我们在一个 IDE 类型环境中编写着色器，其中某些还将这种方式作为初始方式，这样做有它的好处，而我对此也并不排斥，但是，我更喜欢能够运行的实际程序，以便做一些有趣的事情，这就是我学习编程的方式。然后我可以将程序发送给我的母亲、女朋友、伙伴、老板等，向他们展示我的聪明才智。当我们开始掌握新技术时，这是一种强有力的反馈机制。

这是我的学习方法，也是本书（畅销的）前 4 版所采用的模式，在第 5 版我仍然不愿意放弃这种方式，希望你会喜欢我的努力成果。

——**Richard S. Wright, Jr.**

致　谢

感谢 Nick 和 Graham，谢谢你们承担了本版中这么多的工作；谢谢你们对我的代码进行调试，及时发现那些愚蠢的错误而不至于被读者看到，并且尽管在大多数时候都比我更有才智，却让我获得了大部分赞誉。谢谢你，Debra Williams Cauley，在本书新的一版中再次给予我信任，给予我帮助，并且在我没能及时完成稿件时没有将催稿函发到我家——至少没有经常这样做。Songlin，总有一天我会明白“你的”和“你是”、“它的”和“它是”之间的差别。谢谢你让我看起来好像真的通过了八年级，没有使我觉得自己总是像一个笨蛋。Brian Collins 和 Chris “Xenon” Hanson 在审阅本书早期素材时同样做了大量工作，并且发现了很多潜在的尴尬错误。我会永远感激你们，随时来市里，我还欠你们一顿啤酒。

感谢佛塞大学，十多年来让我能够在继续“白天工作”的同时讲授 OpenGL——特别是 Rob Catto，总是在我时常遇到阻碍时放我一马并伸出援手。感谢我在图形学系的好朋友和同伴 Wendy “Kitty” Jones、Kent Ward 和 Nick Bullock，谢谢你们所有精神上和物质上的帮助，不期而至的泰国美食，以及某些时候代替我完成我的工作。

特别感谢软件 Software Bisque（Steve、Tom、Daniel 和 Matt）让我每天都可以使用 OpenGL 做一些“真实”的东西，并且为我提供了人们梦寐以求的可能是最酷的白天（还有晚间）工作。我还要感谢我的家庭，LeeAnne、Sara、Stephen 和 Alex。你们忍受了太多的情绪波动、飘忽不定的关注点和毫无规律的工作日程，更不用提在我需要的时候给我的那些激励了。

最后，感谢苹果公司，没有让我在每一次需要重启计算机来改变操作系统时都要等待“正在安装重要更新，请不要关闭计算机电源”。感谢 AMD/ATI 为我提供了那么酷的新工具，它们确实很有帮助。非常高兴看到你们竭尽全力地支持 OpenGL 标准。

—Richard S. Wright, Jr.

首先感谢 Richard，让我参与创造又一个 OpenGL 出版物里程碑的荣誉之旅。如果没有你的付出和奉献，计算机图形学的学生们就会失去学习 3D 图形的必要工具。和你一起长期共事来支持 3D 图形特别是 OpenGL 是我的荣幸。感谢 Graham 帮助我们将这一版带入最前沿的 OpenGL 3.3。你目光如炬，帮助我避免了许多麻烦，并使这本书一直保持着紧扣主题。谢谢你，Debra Williams Cauley，让我们轻松回到出版流程，并且引导我们这些外行人完成工作。你的耐心是特别的、无私的。感谢 Songlin 专业的目光，对我粗陋的文字进行润色。

如果没有那些为数众多的反馈意见，这本书就不可能这样成功。特别感谢 AMD 的 Mark Young，你一丝不苟地审阅了我的所有书稿，并且提供了宝贵的反馈意见，而你并没有义务这样做。Brian Collins 和 Chris Hanson，你们对于确保素材质量一流和没有错误来说至关重要，感谢你们及时的反馈。

我还要感谢 AMD 和所有杰出的 OpenGL 开发者，你们提供了极大的支持和帮助，促使 OpenGL 3.3 在实际硬件上真正可用，使那些实例变得好用，也让我的工作得以进行。感谢 Mark Young、Mais Alnasser、Ken Wang、Jaakko Konttinen、Murat Balci、Bird Zhang、Zhihong Wang、Frank Li、Erick Zolnowski、Qun Lin、Jesse Zhou、Ethan Wei、Zhaohua Dong，以及其他许多人，你们都做出了杰出的工作。特别感谢 Khronos 小组和所有为保持 OpenGL 作为唯一的真正跨平台 3D API 的先进性、相关性和竞争力而努力的团体。

当然，如果没有家庭和朋友的支持，我就无法完成这项工作。谢谢我的妻子 Anna，这些年你容忍了我含糊混乱的技术术语，同时又在自己的领域治病救人，并且在制药方面有所建树。谢谢你的耐心和支持，没有你我就无法成功。

——**Nicholas Haemel**

感谢我的合作者 Richard 和 Nick，也感谢我们的出版人对我的信任，让我能够勉力地应付这样厚的一本书。和你们共事是一种享受，我希望这仅仅是一个开始。谢谢 Dave，谢谢你说“当然，我喜欢 Graham”。谢谢 Debra，谢谢你在必要时（必要的时候经常出现）敦促我交稿。感谢 Songlin 教会我如何正确地设置文本格式。从我们的技术审稿人 Brian Collins 和 Chris “Xenon” Hanson 那里得到的反馈使我受益匪浅，并且让我确信自己没有说什么愚蠢的话。

我还要衷心感谢 AMD 的同事们。特别地，我要感谢 Mark Young，他甚至在我的稿件送给技术审稿人之前并且还没有成型时就阅读了它们。Mark 还作了大量努力来更新本书附录中的 OpenGL 参考。这真的让我喜出望外——谢谢你！感谢 Jaakko、Murat 和其他在我们为本书实例进行调试时提供了建议和帮助的每一个人。我真的非常享受我们的头脑风暴，你们的参与极具价值，如果没有你们，我所做的一半工作都会不好用。感谢 Bill 和 Nick（再次感谢）帮助我联系 Khronos 和 ARB。Pierre，你是一位了不起的导师。感谢 Suki 使我能够脱离工作内容，来做那么多我想做的事，你为我提供了绝好的机会，我非常感激。

我要感谢一直以来帮助我的每个人：我在 Epson 的老同事，ARB 的同僚——感谢你们如此包容这个只出现了一天的家伙。这些人中很多都是我的竞争对手，我很庆幸我们能够在这么多事情上联手合作。

最后，我要深深地感谢我的家庭。Chris，你真棒，你给了我这么多，我爱你。Jeremy，你太棒了。妈妈、爸爸，谢谢你们创造了我！Barry 和 Phyllis，谢谢你们创造了 Chris！

——**Graham Sellers**

关于作者

Richard S. Wright, Jr. 拥有 15 年以上的 OpenGL 使用经验。他在美国佛罗里达州奥兰多附近的佛塞大学游戏设计学位课程（game design degree program）中教授编程，至今已逾 10 年。现在，Richard 是 Software Bisque 公司的资深工程师，是负责 Seeker（一款 3D 太阳系模拟软件）的技术主管，也是他们的全圆顶剧场天文馆（full dome theater planetarium）产品的产品经理。

多年前在洛克希德曼丁公司的 Real 3D 子公司时，Richard 是正规 OpenGL ARB 的成员，并参与了 OpenGL 1.2 规范和一致性测试。从那时起，Richard 就开始致力于 Windows、Linux、Mac OS X 和各种移动平台上的多维数据库可视化、游戏开发、医学诊断可视化和天文空间模拟。

Richard 最早是在 1978 年上 8 年级时开始在纸质终端上学习编程的。在 16 岁时，他的父母让他用割草得到的钱来买一台计算机而不是汽车，过了不到一年，他就卖出了他的第一款计算机程序（并且这还是一个图形程序！）。当高中毕业时，他的第一份工作是为当地消费教育公司教授编程和计算机文化。他在路易斯维尔大学 Speed 科学院学习电机工程和计算机科学，并且在高年级上到一半时就完成了学业，然后他的职业生涯到了最佳时期并来到了佛罗里达州。作为一个土生土长的肯塔基州路易斯维尔人，他目前和他的妻子、3 个孩子一起居住在佛罗里达州的玛丽湖。在编程和躲避飓风之外的闲暇时间里，Richard 是一个热心的天文爱好者和摄影爱好者。Richard 还是一个 Mac 拥护者，并以此为荣。

Nicholas Haemel 在 OpenGL 得到广泛接受之后不久就开始接触它了，迄今已逾 12 年。他毕业于密尔瓦基工学院（Milwaukee School of Engineering）并获得计算机工程学位，并且开始热爱嵌入式系统、计算机硬件以及让它们运行起来。刚一毕业，他就在 ATI 的 3D 驱动小组中发挥了这些技能，开发图形驱动和新的 GPU。

现在，Nick 是 AMD（高级微设备公司）OpenGL 小组的技术组成员，并且已经成为驱动构架、设计和开发中的重要贡献者。他还领导着工作站 OpenGL 市场的所有行动和项目。在过去的 4 年中，Nick 还参加了体系结构审核委员会（ARB，现在是 Khronos 小组的一部分），并参与了 OpenGL 3.0、3.1、3.2、3.3 和 4.0 规范和相关扩展及 GL 着色语言版本的定义。除了 OpenGL 之外，他还为 OpenGL ES、WebGL 和 EGL 工作组作出了贡献。

Nick 的图形生涯开始于 9 岁那年，也就是他第一次学习使用 Logo Writer 进行 2D 图形编程时。在说服父母购买了一台最新型的 286IBM 兼容机之后，这台计算机马上就成了机械手和其他远程可编程设备的中心控制单元。20 年后，这些被控制的设备则是指甲大小但却包含 20 亿个晶体管的 GPU。Nick 的兴趣还包括商业领导和管理，特别是最近获得威斯康星大学麦迪逊分校（University of Wisconsin–Madison）

的 MBA 之后更是如此。这所学校也是他现在的居住地。在致力于推进未来图形硬件发展的工作之余，Nick 喜欢作为竞技水手、登山队员、滑雪选手、自行车运动员和摄影师进行户外运动。

Graham Sellers 是一个典型的极客（geek）。在他 6 岁生日前夕，家里拥有了第一台计算机（BBC Model B）。在父母通宵用计算机进行编程来播放“生日快乐”之后，他就迷上了它，并且决定要搞清楚它是如何工作的。接下来就是基础编程，然后是装配语言。他第一次真正接触图形是在上世纪 90 年代早期的“demos”，然后是 Glide，最后是上世纪 90 年代晚期的 OpenGL。他拥有英国南安普顿大学的工程硕士学位。

现在，Graham 是 AMD 的 OpenGL 驱动小组中的一名经理。他在 ARB 中代表 AMD，并对核心 OpenGL 规范和很多扩展作出了贡献。在此之前，他曾经是爱普生的一位项目组长，负责实现嵌入式产品的 OpenGL-ES 和 OpenVG 驱动。Graham 在计算机图形和图像处理领域拥有多项专利。在从事相关 OpenGL 工作之余的闲暇时间里，他喜欢进行老视频游戏控制台的分解和反向工程（仅仅是为了解它们如何工作，以及他能利用它们做些什么）。Graham 来自英格兰，目前与他的妻子和儿子一起居住在佛罗里达州的奥兰多。

前　言

欢迎阅读本书第 5 版。在过去十多年的时间里，我们致力于不仅为 OpenGL，也为通用 3D 图形编程提供全世界最好的入门指南。本书既是 OpenGL API 的全面参考书，也是一本能够教会读者如何使用这个强大的 API 来创建绝妙的 3D 可视化、游戏和其他所有类型图形的教程。本书从基础 3D 术语和概念开始，带领读者学习基本图元装配、变换、光照、纹理，并最终带领读者学习使用 OpenGL 着色语言的可编程图形管线的完整功能。

无论读者在 Windows、Mac OS X、Linux，还是在手持游戏设备上进行编程，本书都是开始学习 OpenGL，以及在读者特定平台上最大程度利用它的好工具。本书的主体是以 GLUT 或 FreeGLUT 工具箱为宿主的高度可移植的 C++代码。读者还可以在本书中看到特定操作系统的章节，这些章节展示如何将 OpenGL 写入本地窗口系统。在本书中，我们自始至终努力不去假设读者有多少 3D 图形编程预备知识，这样就促成了这本初学的程序员和经验丰富的程序员开始学习 OpenGL 时都能接受的教程。

第 5 版有哪些更新

本书前几版的读者可能会立刻注意到，这本书变薄了。这是怎么回事？在 OpenGL 3.0 中，某些特性被标记为“不鼓励使用的”，也就是说，这些特性在未来的 OpenGL 版本中可能会被删除。到目前为止，OpenGL 中还没有正式删去任何特性，这在很大程度上是迫于广大开发人员的压力。这样一来，我们在讲解 OpenGL 就有了两种思路，即包括所有最新功能和“不鼓励使用的”功能的完整框架，以及不包括任何“不鼓励使用的”功能的核心框架。鉴于标记“不鼓励使用的”功能的主要目的是推动 OpenGL 标准的发展，本版没有包含任何“不鼓励使用的”功能，而只专注于核心框架。这些核心框架是基于 OpenGL 3.3 的。

我们保留了已经证实非常受欢迎的书后参考资料，不过同样删去了所有“不鼓励使用的”功能。如果读者想实现最新的和具有前瞻性的 OpenGL 程序，那么从这一部分开始是非常好的选择。教程部分的各章内容 95%（或以上）都是新素材。我们不想采用基于“不鼓励使用的”功能的方式，所以就产生了采用全新素材的全新方式。这其中包括本版中关于操作系统特性的各章内容，这些内容几乎是完全重写的。

关于 OpenGL ES 的章节特别加入了在 iPhone 上使用 OpenGL ES 的内容，其中还包括了 iPod 和

iPad，并且本书前面的一些例子也引入了这些设备。这些新加入的内容是非常受欢迎的，因为在本书编写时，再没有一种主流 OpenGL ES 设备能够像它们这样为任何（使用 Mac）用户轻松使用。

在本版中，GLTools 库部分也被明显地加强了。书中归纳的存储着色器(stock shader)能帮助读者在真正开始研究编写自己的着色器前就尽快学会如何使用着色器。另外，本书归纳的轻量级 C++类允许对我们的几何批处理进行管理，并帮助我们创建和操作自己的矩阵堆栈。像过去的 GLU 库一样，这个库也应只被视为一套帮助例程，而不是一个完整的 OpenGL 应用框架。

本书的组织结构

本书共分为 3 部分。第一部分是 OpenGL/3D 图形学基本教程；第二部分涵盖了更加深入的 OpenGL 编程主题；第三部分则涵盖了一些操作系统特定性质，帮助读者在选定的平台上对 OpenGL 进行充分的利用。这 3 部分之后是 3 个附录，包含了其他 OpenGL 优秀参考资源和教程的汇总、一个简短的术语表，以及 OpenGL 核心框架的完整参考。

第一部分　基本概念

在这一部分，读者将学到如何构建一个使用 OpenGL 的程序，如何设置 3D 渲染环境，以及如何创建基本对象和光线并对他们进行着色。然后，我们将深入研究如何使用 OpenGL，并向读者介绍 GLSL，以及如何创建自己的着色器。这些章节是读者认识使用 OpenGL 进行 3D 图形编程很好的方式，并提供关于本书后面将讲到的更多高级性能的基础概念。

第 1 章　3D 图形和 OpenGL 简介　本章是一个针对 3D 图形新手的介绍性章节。它介绍了基本概念和一些通用词汇。

第 2 章　入门指南　在本章，我们向读者提供了关于 OpenGL 是什么、它从何而来以及将如何发展的应用知识。读者将编写自己的第一个使用 OpenGL 的程序，找出需要使用哪些头和库，学习如何设置环境，并发现一些通用惯例如何帮助读者记住 OpenGL 函数调用。本章还会介绍 OpenGL 状态机和错误处理机制。

第 3 章　基础渲染　在本章，我们展示 3D 图形编程的构造块。读者可以大致明白如何告诉计算机用几何图元、着色器在 OpenGL 中创建一个三维对象，建立统一样式和属性。读者也可以学到如何消除隐藏表面和使用模板缓冲区的方法，以及查询 OpenGL 驱动获得实现细节的各种不同方式。

第 4 章　基础变换：初识向量/矩阵　现在我们在虚拟世界创建三维形状，如何使它们移动？如何使自己移动？这些都在这一章学习。本章真正属于 OpenGL 的内容很少，但却澄清了读者继续深入学习所需要了解的概念。

第 5 章　基础纹理　纹理贴图在任何 3D 图形工具箱中都是最有用的特性之一。我们将学到如何将图像缠绕到多边形上，以及如何立即载入和形成多纹理。

第 6 章　跳出“盒子”：非存储着色器　现在我们已经具备 OpenGL 客户端程序设计基础，可以开始在服务器端如何使用 GLSL 编写着色器下功夫了。本章通过一些基于前面所学内容使用存储着色器创建的

例子来介绍这方面内容。

第 7 章　纹理高级知识　学习基础纹理后，在本章我们将学习立方体贴图、3D 纹理，并且只用纹理存储数据。我们还可以学习点精灵和一些其他类型的非可视纹理应用。

第二部分　深入探索

在本书的第二部分，我们将进行更深入的研究。这一部分关于 OpenGL 的内容更加令人振奋，而懂得如何应用这些高级特性将使读者超越业余 3D 玩家的水平。这一部分不仅能够使我们掌握更多的可视效果，而且其中很多内容还是性能导向的。

第 8 章　缓冲区对象：存储尽在掌握　OpenGL 不再支持客户端的数据存储。在本章，读者将学习 OpenGL 中不同类型存储缓冲区的输入输出，包括如何在离屏缓冲区中渲染。

第 9 章　高级缓冲区：超越基础水平　本章将向读者展示如何进一步学习更高水平的缓冲区知识，并介绍一些非常有用但却并不典型的缓冲区格式。

第 10 章　片断操作：管线的终点　如果片段着色器的颜色、深度或其他数据变得不精确时，仍然需要进行很多处理。本章讨论对逐个片段的操作，包括非常有用的模板测试。

第 11 章　高级着色器应用　本章将帮助读者在着色器编程中引入几何着色器，此外还将介绍更多高级着色器管理和诸如统一块（uniform block）等应用模式。

第 12 章　高级几何图形管理　本章是第二部分的最后一章，介绍一些几何与渲染操作的高级管理方法与技巧。OpenGL 中有一些有用的特性可以用来优化大量几何图形的处理过程，以及消除事先不可见的几何图形。最后，实际上 OpenGL 中现在还内建了一些有用的时间特性。

第三部分　特定平台应用

本书的第三部分也是最后一部分，主要是关于各种操作系统接口如何带有和使用 OpenGL 的，而不是关于 OpenGL 本身的。这部分内容游离在“官方”OpenGL 规范之外，来了解 OpenGL 是如何支持和连接 Windows、Mac OS X、Linux 和掌上设备的，例如使用 OpenGL ES 2.0 的 iPhone。

第 13 章　Windows 上的 OpenGL　在本章，我们将学习如何编写真正的使用 OpenGL 的 Windows 程序。我们将学习 Microsoft 的“wiggle”函数，这个函数将 OpenGL 渲染代码与 Windows 设备环境结合起来。

第 14 章　OS X 上的 OpenGL　在本章，我们将学习如何在本地 Mac OS X 应用程序上使用 OpenGL。示例程序将为我们展示如何使用 Xcode 开发环境开始应用 GLUT、Carbon 或 Cocoa。

第 15 章　Linux 上的 OpenGL　本章讨论 GLX，一种用于通过 UNIX 和 Linux 上的 X Window 系统来支持 OpenGL 应用程序的 OpenGL 扩展。我们将学习如何创建和管理 OpenGL 环境，以及如何创建 OpenGL 绘图区域。

第 16 章　OpenGL ES：移动设备上的 OpenGL　本章完全是关于 OpenGL 如何进行精简以适应手持和嵌入式设备的。我们将了解删去了什么、新增了什么，以及如何在仿真环境下运行。本章甚至还将一个桌面示例程序移植到了 iPhone 上。

关于合作网站

这是本书第二次没有搭配 CD-ROM 发行。欢迎进入 Internet 时代！所有源代码都可以从支持网站在线下载。

www.starstonesoftware.com/OpenGL

在这里可以找到所有示例程序的源代码，以及为 Developers Studio（Windows）和 Xcode（Mac OS X）预先建立的项目。对于 Linux 用户，我们也为项目的命令行建立制作了文件，甚至计划公布一些教程，所以请不时地进行核对，即使是在下载完所有源代码之后。

目　录

第一部分　基本概念

第二部分 深入探索

第三部分 特定平台应用