



Khronos小组编写的OpenGL官方权威指南，素有“OpenGL红宝书”美誉

结合OpenGL 4.3的新特性，全方位阐释OpenGL编程的各种技术细节、方法和最佳实践，帮助程序员踏上OpenGL专家之路

PEARSON

华章程序员书库

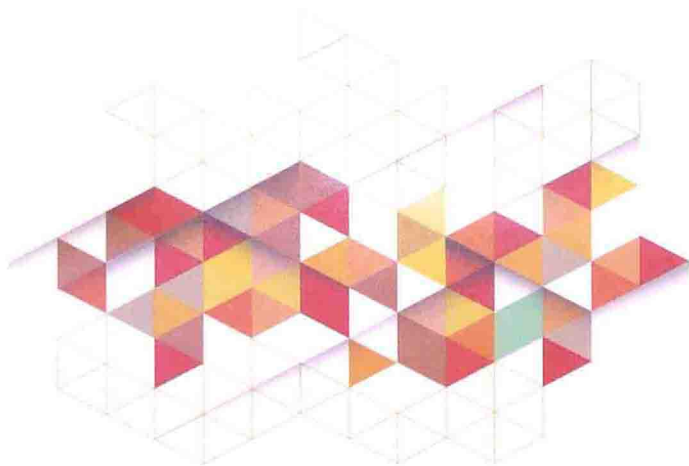
OpenGL Programming Guide  
The Official Guide to Learning OpenGL, Version 4.3  
Eighth Edition

# OpenGL编程指南

(原书第8版)

(美) Dave Shreiner Graham Sellers John Kessenich Bill Licea-Kane 著

王锐 等译



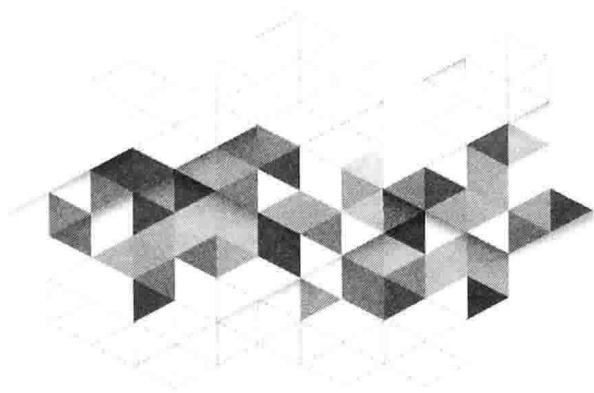
机械工业出版社  
China Machine Press

OpenGL Programming Guide  
The Official Guide to Learning OpenGL, Version 4.3  
Eighth Edition

# OpenGL编程指南

(原书第8版)

(美) Dave Shreiner Graham Sellers John Kessenich Bill Licea-Kane 著  
王锐 等译



机械工业出版社  
China Machine Press

## 图书在版编目 (CIP) 数据

OpenGL 编程指南 (原书第 8 版) / (美) 施莱尔 (Shreiner, D.) 等著; 王锐等译. —北京: 机械工业出版社, 2014.10

(华章程序员书库)

书名原文: OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 4.3, Eighth Edition

ISBN 978-7-111-48113-3

I.O… II. ① 施… ② 王… III. 图形软件 - 指南 IV. TP391.41-62

中国版本图书馆 CIP 数据核字 (2014) 第 226183 号

本书版权登记号: 图字: 01-2013-4455

Authorized translation from the English language edition, entitled OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 4.3, 8E, 978-0-321-77303-6 by Dave Shreiner, Graham Sellers, John Kessenich, Bill Licea-Kane, published by Pearson Education, Inc., Copyright © 2013.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

Chinese simplified language edition published by Pearson Education Asia Ltd., and China Machine Press Copyright © 2014.

本书中文简体字版由 Pearson Education (培生教育出版集团) 授权机械工业出版社在中华人民共和国境内 (不包括中国台湾地区和香港、澳门特别行政区) 独家出版发行。未经出版者书面许可, 不得以任何方式抄袭、复制或节录本书中的任何部分。

本书封底贴有 Pearson Education (培生教育出版集团) 激光防伪标签, 无标签者不得销售。

## OpenGL 编程指南 (原书第 8 版)

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 秦 健

责任校对: 董纪丽

印 刷: 北京市荣盛彩色印刷有限公司

版 次: 2014 年 10 月第 1 版第 1 次印刷

开 本: 186mm × 240mm 1/16

印 张: 41.75

书 号: ISBN 978-7-111-48113-3

定 价: 129.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzsj@hzbook.com

版权所有 · 侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

“这是一本一站式服务的 OpenGL 书籍。它就是我梦寐以求的那种图书。感谢 Dave、Graham、John 和 Bill，感谢你们作出的了不起的贡献。”

——Mike Bailey，俄勒冈州立大学教授

“最近出版的这本红宝书依然遵循了 OpenGL 的伟大传统：不断进化让它拥有了更为强大的力量和效率。第 8 版包含了最前沿的接口标准和新特性的内容，以及对于应用在各行各业的现代 OpenGL 技术的脚踏实地的讲解。红宝书依然是我的公司中所有新员工的必备参考书。还有其他任何一本书可以说的上是必备的指南书吗？它让我喜极而泣，让我觉得无与伦比——我会一遍又一遍地阅读这本书。”

——Bob Kuehne，Blue Newt Software 总裁

“OpenGL 在这 20 年来已经有了巨大的发展。这次的修订版是一本学习使用现代 OpenGL 的实用指南书。现代 OpenGL 侧重于着色器的使用，而这一版的编程指南准确地对应了这一点，它在第 2 章对于着色器进行了深入的叙述。而后继的章节里，它继续深入到方方面面，从纹理到计算着色器。无论你对 OpenGL 了解多少，或者你准备深入到何种程度，只要你准备开始编写 OpenGL 程序，你就一定需要《OpenGL 编程指南》这本手边书。”

——Marc Olano，UMBC 副教授

“如果你正在寻找有关 OpenGL 最新版的编程权威指南，那么你已经找到了。本书的作者深入参与了 OpenGL 4.3 标准的创立，而这本书中恰恰包含了你所需要了解的一切，它将使用一种清晰的、富有逻辑性和见解性的方式，介绍这个行业领先的 API 标准的最新知识。”

——Neil Trevett，Khronos Group 总裁

## 译者序 *The Translator's Words*

OpenGL 发展至今，已经超过了 20 年的时间。作为一个成熟而久负盛名的跨平台的计算机图形应用程序接口规范，它已经被广泛应用于游戏、影视、军事、航空航天、地理、医学、机械设计，以及各类科学数据可视化的领域。而随着网络和移动平台的飞速发展，异军突起的 OpenGL ES 和 WebGL 标准也吸引了大批开发者的眼球，而这两者与 OpenGL 本身同样有着千丝万缕的联系。

OpenGL 支持几乎所有现有的主流操作系统平台，包括 Windows、Mac OS X 以及各种 UNIX 平台。它同时也可以用于几乎所有主流的编程语言环境当中，例如 C/C++、Java、C#、Visual Basic、Python、Perl 等。因此，无可非议地说，OpenGL 应当是目前全球最为广泛学习和使用的图形开发 API 接口，我们几乎可以在全世界任何一台计算机安装的软件当中找到它的身影（当然，在 Windows 平台下总会有 OpenGL 和 DirectX 两类 API 的地位与优劣之争，这又是本书内容之外的另一番故事了）。

而作为 OpenGL 学习的经典书籍，有着“红宝书”之名的本书也已经更新到了第 8 版。这一版的最大特色就是“变革”。是的，这是一本变革之书，它直接与 OpenGL 4.3 版本的内容相贴合，彻底以核心模式的主要函数与着色器的内容为讲解重点。如果你已经读过以前的一些修订版本，并且已经对 `glBegin()`、`glLoadMatrix()`，或者 `display list` 这些函数和名词耳熟能详，那么不要惊讶：从这一版开始，你将再也见不到有关这些内容的介绍。即使你是一位从业数十年的 OpenGL 开发者，从这一刻开始，你恐怕也需要从头来过。

不过无须惊惶，你手头的已经开发了多年的 OpenGL 程序，依然可以在兼容模式下顺利执行。而基于可编程流水线的全新架构和接口，想必也会给有经验的开发者带来更多的思考与创新力。而对于初涉 OpenGL 开发的新人来说，这恰恰是一个好机会，让你们从新的起点出发，把学习和思考的重心放在以着色语言为基础的体系之上，而不是死记硬背那些程序接口，或者苦恼于繁杂的状态切换。

本书内容翔实，章节划分清晰明确，适合各种层次的读者选择性地阅读。不过，作为一个具有颠覆性质的修订版本，这一版的“红宝书”也暴露出内容讲解上的一些问题，部

分章节略显臃肿冗余，而内容的衔接上也不够连贯，这样都可能让没有基础的新人感到无所适从。应当说，这也是本书未来亟待更新和改进的方面。

译者作为基于 OpenGL 的开源 3 维引擎 OpenSceneGraph 的核心开发者，有幸主持了本版的翻译工作，但是因为时间紧迫，译者才疏学浅，因此错漏之处想必很多，敬请读者谅解。如有任何形式的批评或建议，欢迎随时与译者联系。本书的图文内容与之前的修订版本基本不存在关联，书中第 1~6 章，第 8~11 章均由王锐负责翻译与整理，其他参与本书翻译的人员还有：郭华（第 6 章，附录 G、附录 H 和附录 I）、苏明南（第 7 章和附录 D）、张静（第 12 章）、王凯（附录 A）、陈节（附录 B）、龙海鹰（附录 C）、毕玉玲（附录 E）。感谢他们的辛苦付出，也感谢机械工业出版社的编辑们的信任与帮助！

## 前言 *Preface*

OpenGL 图形系统是图形硬件的一种软件接口（GL 表示 Graphics Library，即图形库）。它使得用户可以创建交互式的程序以产生运动的 3 维对象的颜色图像。通过 OpenGL，我们可以使用计算机图形学技术来产生逼真的图像，或者通过一些虚构的方式产生虚拟的图像。这本指南将告诉你如何使用 OpenGL 图形系统进行编程，得到你所期望的视觉效果。

### 本书的主要内容

本书包含以下一些章节：

- 第 1 章对 OpenGL 可以完成的工作进行了概览。它还提供了一个简单的 OpenGL 程序并解释了一些本质性的编程细节，它们可能会用于后继的章节中。
- 第 2 章讨论了 OpenGL 中最主要的特性——可编程着色器，并介绍了它们在应用程序中的初始化和使用方法。
- 第 3 章介绍了使用 OpenGL 进行几何体绘制的各种方法，以及一些可以让渲染更为高效的优化手段。
- 第 4 章解释了 OpenGL 对于颜色的处理过程，包括像素的处理、缓存的管理，以及像素处理相关的渲染技术。
- 第 5 章给出了将 3 维场景在一个 2 维计算机屏幕上表现的操作细节，包括各种几何投影类型的数学原理和着色器操作。
- 第 6 章讨论了将几何模型与图像结合来创建真实的、高质量的 3 维模型的方法。
- 第 7 章介绍了计算机图形的光照效果模拟方法，主要是这类方法在可编程着色器中的实现。
- 第 8 章介绍了使用可编程着色器生成纹理和其他表面效果的方法细节，从而增强真实感和其他的渲染特效。
- 第 9 章解释了 OpenGL 管理和细分几何表面的着色器功能。

- 第 10 章介绍了一个在 OpenGL 渲染流水线中使用着色器进行几何体图元修改的特别技术。
- 第 11 章介绍了使用 OpenGL 帧缓存和缓存内存实现高级渲染技术和非图形学应用的相关方法。
- 第 12 章介绍了最新的着色器阶段，将通用计算的方法融合到 OpenGL 的渲染流水线当中。

此外，我们也提供了一系列作为参考的附录内容。

- 附录 A 介绍了 OpenGL Utility Toolkit 这个专用于窗口系统操作的库。GLUT 是可移植的，它可以用来实现更简短也更加可读的代码案例。
- 附录 B 介绍了 OpenGL 体系中的其他 API，包括用于嵌入式和移动平台系统的 OpenGL ES，以及用于 Web 浏览器内的交互式 3D 应用程序的 WebGL。
- 附录 C 提供了有关 OpenGL 着色语言的详细参考文档。
- 附录 D 列出了 OpenGL 维护的所有状态变量，并介绍了获取其数值的方法。
- 附录 E 介绍了与矩阵变换相关的一些数学方法。
- 附录 F 介绍了不同的窗口系统相关的各种库，它们提供了各种绑定例程，以支持 OpenGL 渲染到本地窗口当中。
- 附录 G 对于 OpenGL 中所用到的浮点数格式做出了概述。
- 附录 H 介绍了 OpenGL 中最新的调试特性。
- 附录 I 给出了有关 uniform 缓存的使用的参考文档，其中使用了 OpenGL 定义的标准内存布局。

## 本版新增内容

本书的内容是颠覆性的！对于那些已经阅读过本书以前版本的读者来说，这一版完全根据 OpenGL 应用程序开发的最新方法和技术进行了重写。本书将经典红宝书的以函数为中心的方法，与《OpenGL 着色语言》(通常也称作“橘皮书”)一书中的着色技术进行了融合。

在这一版中，作者团队中也纳入了 OpenGL 开发的一些主要贡献者，以及 OpenGL 着色语言标准的编者。因此，这一版将会涵盖 OpenGL 的最新版本，也就是版本 4.3，其中包括了计算着色器的内容。本书还介绍了可编程渲染管线的每个阶段。我们衷心地希望你能够从本书中找到实用的以及有教育价值的内容。

## 你需要在阅读本书之前掌握的知识

本书假设你已经了解了使用 C 语言进行编程的方法(我们将使用少量的 C++ 程序，不过你应该会比较容易理解它们)，并且具有一定的数学背景(几何、三角学、线性代数、微



积分, 以及微分几何)。即使你对于计算机图形学技术没有太多的经验或者一无所知, 你也可以学习和理解本书中讨论的大部分内容。当然, 计算机图形学是一个不断延展的学科, 因此你也许还需要阅读以下的补充内容来提升自己的知识。

□《Computer Graphics: Principles and Practice》第3版, John F. Hughes 等著 (Addison-Wesley, 2013): 这本书是有关计算机图形学的一本百科全书。它包含了大量有价值的信息, 不过在阅读它之前, 你最好已经对这门学科有了一定的了解。

□《3D Computer Graphics》, Andrew S. Glassner 著 (The Lyons Press, 1994): 这本书是有关计算机图形学的非技术性的、适度的介绍。它注重于可以实现的可视化效果本身, 而不是实现这些效果所需的技术。

另一个可以有组织地进行系统学习的地方就是 OpenGL 网站。该网站包含了软件、示例程序、文档、FAQ、讨论版, 以及新闻页面。如果你想要搜索 OpenGL 相关问题的答案, 那么这里是一个好的开始:

<http://www.opengl.org/>

此外, OpenGL 的官方网站中还包含了 OpenGL 最新版本对应的所有函数和着色语言语法的完整文档。这些网页内容完整地涵盖了《OpenGL Reference Manual》一书的内容, 后者由 OpenGL Architecture Review Board 和 Addison-Wesley 出版。

OpenGL 是一个与硬件密切相关的编程接口标准, 我们可能会在某一类特定的硬件上使用一个特定的 OpenGL 实现。本书将会介绍如何使用任意的 OpenGL 实现进行编程。但是, 因为这些实现之间会存在细微的差异——包括性能上的差异, 以及额外的特性支持——你可能需要阅读自己所用的特定设备实现所对应的补充文档。此外, 某个特定实现的供应商网站上, 也可能也会提供一些 OpenGL 相关的实用工具、工具包、编程和调试支持、窗口组件、示例代码, 以及示例程序。

## 如何获取示例代码

本书包含很多示例程序, 它们演示了特定 OpenGL 编程技术的用法。本书的读者群体在计算机图形学和 OpenGL 方面可能有着巨大的经验差异, 有的人是新手, 而有的人是多年的老手, 因此这些章节里给出的案例都会使用最简单的方法去实现一个特定的渲染形式, 并且全部使用 OpenGL 4.3 版本的接口。这样的做法主要是为了确保那些刚开始学习 OpenGL 的读者也能够顺利地阅读相关的内容。对于那些已经有了足够的经验, 只是希望了解最新的 API 特性实现的读者, 我们首先感谢你能够耐心阅读本书前面的内容, 之后你可以访问我们的网站:

<http://www.opengl-redbook.com/>

在这里你将会找到本书中所有示例的源代码, 它们均使用最新的特性进行实现, 而后文的讨论中也会涉及从一个 OpenGL 版本移植到另一个版本所需的修改。

本书中所有的程序都使用了 OpenGL Utility Toolkit ( GLUT ), 它的原作者为 Mark Kilgard。在这一版中, 我们将使用 GLUT 接口的开源版本, 它来自 freeglut 工程。这个工程对 Mark 的原始工程 ( 在作者的著作《 OpenGL Programming for the X Window System 》中进行了详细介绍, Addison-Wesley, 1997 ) 进行了加强。你可以在下面的地址里找到开源的工程页面:

<http://freeglut.sourceforge.net/>

你可以在这个网站中找到相应的代码和二进制程序。

本书还介绍了有关 GLUT 库的信息。可以在 OpenGL 网站的资源页面找到更多帮助你学习和使用 OpenGL 与 GLUT 的资源。

<http://www.opengl.org/resources/>

OpenGL 的很多实现也包含了一些系统相关的代码示例。这些源代码可能是你实现程序时最好的资源, 因为它们已经针对系统进行了优化。你可以阅读与自己的系统相关的 OpenGL 文档来了解如何获取这些代码示例。

## 勘误

遗憾的是, 这本书中一定也存在着错误。此外, 即使在本书出版的期间, OpenGL 也是不断更新的: 有一些错误被修正, 并且标准文档中也做出了澄清, 同时还有新的标准被发布。我们将在网站 <http://www.opengl-redbook.com/> 上维护一个错误和更新列表, 同时我们也会提供一些功能让用户提交自己发现的错误。如果你发现了本书中的错误, 我们首先向你郑重道歉, 并且非常感谢你的报告。我们将尽快对其进行更正。

# 目录 Contents

推荐语  
译者序  
前言

<b>第 1 章 OpenGL 概述</b> .....	1
1.1 什么是 OpenGL	1
1.2 初识 OpenGL 程序	2
1.3 OpenGL 语法	6
1.4 OpenGL 渲染管线	7
1.4.1 准备向 OpenGL 传输数据	8
1.4.2 将数据传输到 OpenGL	8
1.4.3 顶点着色	9
1.4.4 细分着色	9
1.4.5 几何着色	9
1.4.6 图元装配	9
1.4.7 剪切	9
1.4.8 光栅化	9
1.4.9 片元着色	10
1.4.10 逐片元的操作	10
1.5 第一个程序：深入分析	10
1.5.1 进入 main() 函数	10
1.5.2 OpenGL 的初始化过程	12
1.5.3 第一次使用 OpenGL 进行渲染	21

<b>第 2 章 着色器基础</b> .....	25
2.1 着色器与 OpenGL	26
2.2 OpenGL 的可编程管线	26
2.3 OpenGL 着色语言概述	28
2.3.1 使用 GLSL 构建着色器	28
2.3.2 存储限制符	34
2.3.3 语句	37
2.3.4 计算的不变性	41
2.3.5 着色器的预处理器	43
2.3.6 编译器的控制	45
2.3.7 全局着色器编译选项	45
2.4 数据块接口	46
2.4.1 uniform 块	46
2.4.2 指定着色器中的 uniform 块	47
2.4.3 从应用程序中访问 uniform 块	48
2.4.4 buffer 块	53
2.4.5 in/out 块	54
2.5 着色器的编译	54
2.5.1 我们的 LoadShaders() 函数	58
2.6 着色器子程序	58
2.6.1 GLSL 的子程序设置	59
2.6.2 选择着色器子程序	60

2.7 独立的着色器对象.....	62	4.4 多重采样.....	115
<b>第3章 OpenGL 绘制方式.....</b>	<b>64</b>	4.4.1 采样着色.....	116
3.1 OpenGL 图元.....	64	4.5 片元的测试与操作.....	117
3.1.1 点.....	65	4.5.1 剪切测试.....	118
3.1.2 线、条带与循环线.....	66	4.5.2 多重采样的片元操作.....	118
3.1.3 三角形、条带与扇面.....	66	4.5.3 模板测试.....	119
3.2 OpenGL 缓存数据.....	69	4.5.4 模板的例子.....	120
3.2.1 创建与分配缓存.....	69	4.5.5 深度测试.....	122
3.2.2 向缓存输入和输出数据.....	71	4.5.6 融混.....	124
3.2.3 访问缓存的内容.....	75	4.5.7 融混参数.....	125
3.2.4 丢弃缓存数据.....	80	4.5.8 控制融混的参数.....	125
3.3 顶点规范.....	80	4.5.9 融混方程.....	127
3.3.1 深入讨论 VertexAttrib- Pointer.....	81	4.5.10 抖动.....	128
3.3.2 静态顶点属性的规范.....	84	4.5.11 逻辑操作.....	128
3.4 OpenGL 的绘制命令.....	86	4.5.12 遮挡查询.....	129
3.4.1 图元的重启动.....	92	4.5.13 条件渲染.....	132
3.5 多实例渲染.....	96	4.6 逐图元的反走样.....	133
3.5.1 多实例的顶点属性.....	97	4.6.1 线段的反走样.....	134
3.5.2 在着色器中使用实例 计数器.....	102	4.6.2 多边形的反走样.....	135
3.5.3 多实例方法的回顾.....	104	4.7 帧缓存对象.....	135
<b>第4章 颜色、像素和帧缓存.....</b>	<b>105</b>	4.7.1 渲染缓存.....	137
4.1 基本颜色理论.....	106	4.7.2 创建渲染缓存的存储 空间.....	138
4.2 缓存及其用途.....	107	4.7.3 帧缓存附件.....	140
4.2.1 缓存的清除.....	109	4.7.4 帧缓存的完整性.....	142
4.2.2 缓存的掩码.....	110	4.7.5 帧缓存的无效化.....	144
4.3 颜色与 OpenGL.....	110	4.8 多重渲染缓存的同步写入.....	145
4.3.1 颜色的表达与 OpenGL.....	111	4.8.1 选择颜色缓存来进行读写 操作.....	146
4.3.2 顶点颜色.....	112	4.8.2 双源融混.....	148
4.3.3 光栅化.....	114	4.9 像素数据的读取和拷贝.....	150
		4.10 拷贝像素矩形.....	152

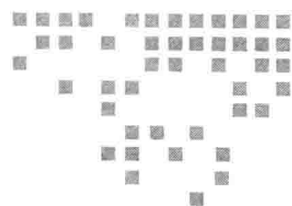
<b>第 5 章 视口变换、剪切与反馈</b> .....	153	6.5.4 从文件加载图像	212
5.1 观察视图	154	6.5.5 查询纹理数据	215
5.1.1 视图模型	154	6.5.6 纹理数据布局	215
5.1.2 相机模型	154	6.6 采样器对象	219
5.1.3 正交视图模型	157	6.6.1 采样器参数	220
5.2 用户变换	158	6.7 使用纹理	221
5.2.1 矩阵乘法的回顾	159	6.7.1 纹理坐标	223
5.2.2 齐次坐标	161	6.7.2 组织纹理数据	226
5.2.3 线性变换与矩阵	163	6.7.3 使用多重纹理	227
5.2.4 法线变换	173	6.8 复杂纹理类型	229
5.2.5 OpenGL 矩阵	174	6.8.1 3 维纹理	229
5.3 OpenGL 变换	177	6.8.2 数组纹理	231
5.3.1 高级技巧: 用户剪切	178	6.8.3 立方体映射纹理	231
5.4 transform feedback	179	6.8.4 阴影采样器	237
5.4.1 transform feedback 对象	180	6.8.5 深度模板纹理	238
5.4.2 transform feedback 缓存	181	6.8.6 缓存纹理	238
5.4.3 配置 transform feedback 的变量	183	6.9 纹理视图	240
5.4.4 transform feedback 的启动 和停止	187	6.10 压缩纹理	243
5.4.5 transform feedback 的示例: 粒子系统	189	6.11 滤波	245
<b>第 6 章 纹理</b> .....	195	6.11.1 线性滤波	245
6.1 纹理映射	196	6.11.2 使用和生成 mipmap	247
6.2 基本纹理类型	197	6.11.3 计算 mipmap 级别	251
6.3 创建和初始化纹理	198	6.11.4 mipmap 细节层次控制	252
6.3.1 纹理格式	202	6.12 高级纹理查询函数	252
6.4 代理纹理	207	6.12.1 显式细节层次	252
6.5 设置纹理数据	208	6.12.2 显式梯度设置	253
6.5.1 显式设置纹理数据	208	6.12.3 偏移后的纹理获取	253
6.5.2 使用 Pixel Unpack 缓存	210	6.12.4 投影纹理	254
6.5.3 从帧缓存拷贝数据	211	6.12.5 着色器中的纹理查询	254
		6.12.6 收集纹素	256
		6.12.7 合并特殊函数	256
		6.13 点精灵	257
		6.13.1 带纹理的点精灵	257

6.13.2 控制点的外观	259	8.2.4 法线贴图	326
6.14 渲染到纹理贴图	260	8.3 程序式纹理的反走样	326
6.14.1 丢弃已渲染数据	263	8.3.1 走样的来源	327
6.15 本章总结	264	8.3.2 避免走样问题	328
6.15.1 纹理回顾	264	8.3.3 提高分辨率	329
6.15.2 纹理的最好实践	265	8.3.4 高频率的反走样	330
<b>第7章 光照与阴影</b>	<b>266</b>	8.3.5 频率截断	337
7.1 光照介绍	267	8.3.6 程序式反走样的总结	339
7.2 经典光照模型	267	8.4 噪声	339
7.2.1 不同光源类型的片元着色器	268	8.4.1 噪声的定义	341
7.2.2 将计算移到顶点着色器	277	8.4.2 噪声纹理	345
7.2.3 多个光源和材质	279	8.4.3 权衡	348
7.2.4 光照坐标系统	285	8.4.4 一个简单的噪声着色器	349
7.2.5 经典光照模型的局限	285	8.4.5 湍流	351
7.3 光照模型进阶	286	8.4.6 大理石	353
7.3.1 半球光照	286	8.4.7 花岗岩	353
7.3.2 基于图像的光照	289	8.4.8 木纹	354
7.3.3 球面光照	293	8.4.9 噪声的总结	357
7.4 阴影映射	296	8.5 更多信息	357
7.4.1 创建一张阴影贴图	297	<b>第9章 细分着色器</b>	<b>359</b>
7.4.2 使用阴影贴图	299	9.1 细分着色器	359
<b>第8章 程序式纹理</b>	<b>303</b>	9.2 细分面片	360
8.1 程序式纹理	303	9.3 细分控制着色器	361
8.1.1 规则的花纹	305	9.3.1 生成输出面片的顶点	362
8.1.2 玩具球	311	9.3.2 细分控制着色器的变量	362
8.1.3 晶格	318	9.3.3 细分的控制	363
8.1.4 程序式着色方法的总结	319	9.4 细分计算着色器	367
8.2 凹凸贴图映射	319	9.4.1 设置图元生成域	368
8.2.1 应用程序设置	321	9.4.2 设置生成图元的面朝向	368
8.2.2 顶点着色器	323	9.4.3 设置细分坐标的间隔	368
8.2.3 片元着色器	324	9.4.4 更多的细分计算着色器 layout 选项	368

9.4.5 设置顶点的位置 .....	369	<b>第 11 章 内存</b> .....	420
9.4.6 细分计算着色器的变量 .....	369	11.1 使用纹理存储通用数据 .....	420
9.5 细分实例：茶壶 .....	370	11.1.1 将纹理绑定到图像单元 .....	425
9.5.1 处理面片输入顶点 .....	370	11.1.2 图像数据的读取和写入 .....	427
9.5.2 计算茶壶的细分坐标 .....	371	11.2 着色器存储缓存对象 .....	430
9.6 更多的细分技术 .....	373	11.2.1 写入结构化数据 .....	431
9.6.1 视口相关的细分 .....	373	11.3 原子操作和同步 .....	431
9.6.2 细分的共享边与裂缝 .....	375	11.3.1 图像的原子操作 .....	431
9.6.3 置换贴图映射 .....	376	11.3.2 缓存的原子操作 .....	439
<b>第 10 章 几何着色器</b> .....	377	11.3.3 同步对象 .....	440
10.1 创建几何着色器 .....	378	11.3.4 图像限定符和屏障 .....	444
10.2 几何着色器的输入和输出 .....	380	11.3.5 高性能的原子计数器 .....	452
10.2.1 几何着色器的输入 .....	380	11.4 示例 .....	455
10.2.2 特殊的几何着色器图元 .....	383	11.4.1 顺序无关的透明 .....	455
10.2.3 几何着色器的输出 .....	387	<b>第 12 章 计算着色器</b> .....	466
10.3 产生图元 .....	389	12.1 概述 .....	466
10.3.1 几何体的裁减 .....	389	12.2 工作组及其执行 .....	467
10.3.2 几何体的扩充 .....	390	12.2.1 知道工作组的位置 .....	471
10.4 transform feedback 高级篇 .....	394	12.3 通信与同步 .....	472
10.4.1 多重输出流 .....	395	12.3.1 通信 .....	473
10.4.2 图元查询 .....	399	12.3.2 同步 .....	474
10.4.3 使用 transform feedback 的结果 .....	400	12.4 示例 .....	475
10.5 几何着色器的多实例化 .....	408	12.4.1 物理模拟 .....	476
10.6 多视口与分层渲染 .....	409	12.4.2 图像处理 .....	481
10.6.1 视口索引 .....	409	12.5 本章总结 .....	485
10.6.2 分层渲染 .....	414	12.5.1 计算着色器回顾 .....	485
10.7 本章小结 .....	417	12.5.2 计算着色器的最佳实践 .....	485
10.7.1 几何着色器回顾 .....	417	<b>附录 A GLUT 基础知识</b> .....	487
10.7.2 几何着色器的经验谈 .....	418	<b>附录 B OpenGL ES 与 WebGL</b> .....	493

附录 C 内置 GLSL 变量与函数 .....	504	的浮点格式 .....	612
附录 D 状态变量 .....	552	附录 H OpenGL 程序的调试与 优化 .....	618
附录 E 齐次坐标与变换矩阵 .....	591	附录 I 缓存对象的布局 .....	632
附录 F OpenGL 与窗口系统 .....	596	术语表 .....	635
附录 G 纹理、帧缓存与渲染缓存			





# OpenGL 概述

## 本章目标

阅读完本章内容之后，你将会具备以下能力：

- 描述 OpenGL 的目的，它在创建计算机图像时，能够做什么，不能做什么。
- 了解一个 OpenGL 程序的通用结构。
- 列举出 OpenGL 渲染管线中的多个着色阶段。

这一章将对 OpenGL 做一个大概的阐述。本章主要包含以下几节：

- 1.1 节将解释 OpenGL 的含义，它可以做到、不能做到的事情，以及它的工作方式。
- 1.2 节将展示一个 OpenGL 程序的结构和表现形式。
- 1.3 节介绍 OpenGL 所使用命令的命名格式。
- 1.4 节介绍 OpenGL 创建图像的整体处理管线过程。
- 1.5 节将重新剖析之前的 OpenGL 示例程序，并且对程序的每个部分提供更详尽的解释说明。

## 1.1 什么是 OpenGL

OpenGL 是一种应用程序编程接口 (Application Programming Interface, API)，它是一种可以对图形硬件设备特性进行访问的软件库。OpenGL 库的 4.3 版本 (即本书所使用的版本) 包含了超过 500 个不同的命令，可以用于设置所需的对象、图像和操作，以便开发交互式的 3 维计算机图形应用程序。

OpenGL 被设计为一个现代化的、硬件无关的接口，因此我们可以在不考虑计算机操作