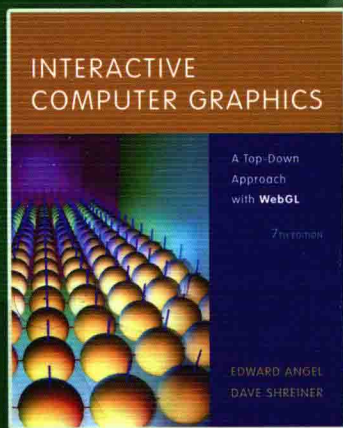


交互式计算机图形学

——基于WebGL的自顶向下方法

(第七版)

Interactive Computer Graphics
A Top-Down Approach with WebGL, Seventh Edition



[美] Edward Angel 著
Dave Shreiner

张荣华 姜丽梅 邵绪强 马玉梅 等译
李继荣 审校



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

国外计算机科学教材系列

交互式计算机图形学

——基于 WebGL 的自顶向下方法

(第七版)

Interactive Computer Graphics
A Top-Down Approach with WebGL
Seventh Edition

[美] Edward Angel Dave Shreiner 著

张荣华 姜丽梅 邵绪强 马玉梅 等译

李继荣 审校

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书采用自顶向下的方法并辅以面向编程的方式,基于现代可编程 GPU 的着色器编程,将 JavaScript 和 OpenGL ES 2.0 结合起来,利用 WebGL 技术编写可以在当前大多数浏览器中运行的示例,同时系统地介绍了现代计算机图形学的核心概念、原理和方法。本书是作者多年来教学与科研工作的总结,涵盖了交互式图形编程、三维可编程绘制流水线、变换与观察、光照与着色、曲线曲面建模等基本的计算机图形学内容以及离散技术、层级建模、过程建模、光线跟踪、并行绘制和体绘制等高级内容。为了方便读者进一步深入学习和研究,本书在每章后面提供了相关的建议阅读资料。

本书既可作为有一定 JavaScript、C、C++ 语言或者 Java 编程基础的计算机及相关专业高年级本科生或研究生的计算机图形学教材,也适合作为相关技术人员的参考书。

Authorized translation from the English language edition, entitled Interactive Computer Graphics: A Top-Down Approach with WebGL, Seventh Edition, 9780133574845 by Edward Angel and Dave Shreiner, published by Pearson Education, Inc., publishing as Addison-Wesley. Copyright © 2015 Pearson Higher Education Inc.

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 PUBLISHING HOUSE OF ELECTRONICS INDUSTRY Copyright © 2016.

本书中文简体字版专有出版权由 Pearson Education(培生教育出版集团)授予电子工业出版社。未经出版者预先书面许可,不得以任何方式复制或抄袭本书的任何部分。

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

版权贸易合同登记号 图字:01-2014-5640

图书在版编目(CIP)数据

交互式计算机图形学:基于 WebGL 的自顶向下方法:第 7 版/(美)安杰尔(Angel,E.), (美)斯赖纳(Shreiner,D.)著;张荣华等译. —北京:电子工业出版社,2016.4

书名原文:Interactive Computer Graphics: A Top-Down Approach with WebGL, Seventh Edition

国外计算机科学教材系列

ISBN 978-7-121-27604-0

I. ①交… II. ①安… ②斯… ③张… III. ①计算机图形学—高等学校—教材 IV. ①TP391.41

中国版本图书馆 CIP 数据核字(2015)第 277710 号

策划编辑:冯小贝

责任编辑:李秦华

印 刷:涿州市京南印刷厂

装 订:涿州市京南印刷厂

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:787×1092 1/16 印张:33 字数:1022 千字 彩插:4

版 次:2009 年 7 月第 1 版(原著第 5 版)

2016 年 4 月第 3 版(原著第 7 版)

印 次:2016 年 4 月第 1 次印刷

定 价:89.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010)88254888。

质量投诉请发邮件至 zlt@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010)88258888。

译者序

计算机图形学(Computer Graphics)产生于20世纪60年代,是研究图形的计算机生成、处理和显示的一门学科。目前计算机图形学已经渗透到工业生产和社会活动的各个领域,从飞机、汽车的计算机辅助设计到飞行模拟器,从视频游戏到数字电影,从天气预报到地质勘探,无一不应用计算机图形生成技术。因此,以图形人机接口和可视化技术为代表的计算机图形学已成为计算机学科中最活跃的分支之一。

在过去十几年,GPU得到了迅速发展,三维图形程序也经历了从运行速度的提高到拥有高质量图形表现力的转变,图形绘制系统性能的增长速度远远超过了摩尔定律,GPU的可编程性也不断增强。对于现代的OpenGL应用程序,所有的绘制都是通过可编程GPU的顶点着色器和片元着色器来控制的,因此新的可编程结构为三维图形应用程序开发人员提供了极大的灵活性。目前桌面版OpenGL的最新版本为OpenGL 4.3。其中,OpenGL 2.0的出现是OpenGL的重大进步,因为它引入了OpenGL着色语言,允许应用程序编程人员编写他们自己的着色器程序并且充分利用GPU的巨大能力。随着移动终端和互联网的广泛应用,OpenGL的变体——OpenGL ES和WebGL陆续衍生出来。OpenGL ES 2.0基于OpenGL 2.0,在嵌入式设备的应用中(如智能手机、平板电脑和游戏机等)获得了良好的图形表现。但是桌面版OpenGL和简化版的OpenGL ES都被设计成在本地运行,不能充分利用Web的优势。2011年Khronos Group发布的第一个WebGL版本源自OpenGL ES 2.0。WebGL是运行在当前大多数Web浏览器上的OpenGL ES 2.0的JavaScript接口,可以为HTML5 Canvas提供硬件3D加速渲染,这样我们可以借助本地的图形硬件在几乎所有最新的浏览器上更流畅地展示3D场景和模型,而不需要额外的专用插件或库。随着GPU和JavaScript引擎的速度与复杂性的日益增加,我们希望更多图形应用程序都可以使用WebGL开发或者移植到WebGL中。

目前国外许多大学都已经将基于可编程GPU的高级着色语言融入计算机图形学的教学与实践中,如University of New Mexico、Harvard、MIT、Caltech、Stanford、Purdue、Brown University、University of Illinois、University of Texas, Austin,等等。因此,随着目前可编程GPU的快速发展,为了切实提高计算机图形学课程的教学质量与教学水平,更好地与国际计算机图形学教育及科研实践接轨,逐渐在我国高校的计算机图形学课程中引入基于可编程GPU绘制流水线及面向着色器编程的新型教学模式已是当务之急。译者多年从事计算机图形学的教学与研究,深知选择这样一本“基础性与先进性、理论性与应用性、科学性与通俗性”相结合且能满足目前计算机图形学新型教学模式教材的重要性。目前在国内外很少有基于这种新型教学模式的计算机图形学书籍,而本书第七版反映了计算机图形学在图形硬件设备和图形绘制标准等方面的最新发展现状。相比于前面六个版本,第七版基于可编程GPU的WebGL着色器编程来介绍计算机图形学的相关原理、方法和技术,编写的三维图形示例能运行于最新的Web浏览器且不需要任何额外的插件或库。因此,本书的第七版能很好地满足新型教学模式的需求。与桌面版OpenGL相比,WebGL更适合讲授计算机图形学,也是培养学生基于现代GPU的可编程计算思维的一本非常难得的优秀教材。

本书的主要特色如下所述。

内容丰富而又系统,基础性内容和拓展性主题相结合。本书涉及了现代计算机图形学课程

几乎所有的主要内容，涵盖了交互式图形编程、三维可编程绘制流水线、变换与观察、光照与明暗绘制、曲线曲面建模等基本的计算机图形学内容以及离散技术、层级建模、过程建模、光线跟踪、并行绘制和体绘制等高级内容。

内容编排和组织独具特色，综合了知识性和实用性。本书采用了“自顶向下”和面向着色器编程的方法，使学生能既使用 WebGL 和 JavaScript 迅速地编写自己的交互式三维 Web 图形应用程序，又能系统地掌握计算机图形学的基本原理、方法和技术。

本书作者结合自己在计算机图形学领域多年的教学和科研成果，为本书提供了大量的经典示例程序、丰富的插图、彩图和习题，并在每章后面提供了最新的建议阅读资料，为进一步加深和扩展读者对有关内容的理解提供了有力支持。

本书既不乏经典理论，又侧重近几年计算机图形学发展的最新成果。本书第七版中新增的某些技术是最近的文献中出现的新技术。为了及时反映计算机图形学的发展现状，本书对第六版中的某些章节进行了调整，并增删了相关内容：

将第六版中有关输入和交互的内容集中起来作为独立的章节，即本书的第 3 章；增加了有关离屏绘制、绘制到纹理、基于 Agent 建模等内容。

所有的应用程序采用 WebGL 技术标准，WebGL 应用程序包含三种语言：HTML5、JavaScript 和 OpenGL 着色器语言，虽然第七版中的许多示例程序的输出结果与以前的版本相同，但代码完全不同。

包含了一个高效的矩阵-向量工具包。

本书原版是国外诸多大学采用的一本经典的计算机图形学教材，译者甚感荣幸再次承接了本书最新版本的翻译工作。为了确保译文的质量，译者和校对人员花费了大量时间对译文进行认真的校对和统稿。对于原文中显而易见的错误，译者在译文中直接进行了更正，而对于原文中可能有误的地方，译文中均以“译者注”的形式做了标注，以供读者参照。

本书的翻译得到了很多人的帮助与参与，在此衷心感谢为本书翻译付出努力的每一个人！参加本书翻译的人员主要有：张荣华、姜丽梅、邵绪强、马玉梅、李继荣、王蓝婧，王新颖也参与了本书的翻译和校对工作，张荣华和李继荣参与了全书的统稿与审校。在此感谢本书第七版的所有译者！

译者在翻译过程中虽然力求准确地反映原著内容，但由于自身的知识局限性，译文中难免有不妥之处，谨向原书作者和读者表示歉意，并敬请读者批评指正。

华北电力大学计算机系
张荣华
2016 年 3 月于保定

前 言

本书是计算机图形学入门教材，重点介绍应用程序的编写。本书于1997年首次出版，第一版在使用 OpenGL 和自顶向下的方法上多多少少是一场革命。在接下来的16年中，发行了六个版本，大多数计算机图形学概论课以及几乎所有的教科书都采用了本书的方法。

本书的第六版反映了主要的图形硬件升级引起的重大图形软件变化。特别地，第六版是完全基于着色器的，可以使读者创建完全利用现代 GPU 性能的应用程序。我们注意到这些改变也被包括在 OpenGL ES 2.0 中，可以用来为嵌入式系统和便携式设备开发应用程序，例如手机和平板电脑，以及 WebGL 的 JavaScript 实现。当初，通过 HTML5 支持 WebGL 的 Web 浏览器出现的时候，我们并没有预料到会对 WebGL 产生如此大的兴趣。

当我们继续编写书籍，讲授 SIGGRAPH 课程以及从事其他图形学相关活动的时候，我们开始意识到对 WebGL 逐渐产生的兴趣。WebGL 的应用如今非常普遍，包括最近智能手机上的一些程序。虽然 WebGL 缺少了一些最新版 OpenGL 的高级功能，但是其与 HTML5 的整合开辟了大量新的应用领域。作为一个额外的好处，我们发现 WebGL 比桌面版 OpenGL 更适合讲授计算机图形学。因此，我们决定使用 WebGL 作为编程语言编写本书的第七版。我们相信第七版也会像其他版本一样带来巨大变革。

第七版新增内容

- 全书使用了 WebGL
- 所有的代码使用 JavaScript 编写
- 所有的代码在近期的 Web 浏览器上运行
- 增加了一章关于交互的内容
- 增加了关于绘制到纹理的内容
- 增加了关于显示网格的内容
- 包含了一个高效的矩阵-向量工具包
- 增加了基于 Agent 建模的介绍

自顶向下的方法

图形学的新进展以及本书前六个版本的成功一直鞭策着我们坚持采用自顶向下的、面向程序设计的方法来介绍计算机图形学。虽然许多计算机科学与工程系现在开设了多门计算机图形学课程，但是大多数学生只选修一门。在学生们学习了程序设计、数据结构、算法、软件工程和基础数学以后给他们安排一门图形学课程，应允许教师以内容充实并且有趣的方式按上述已学课程来组织计算机图形学课程，我们希望这些学生学习这门课程时尽可能早地编写出三维应用程序。底层的算法，例如绘制线段或者填充多边形，可以在学生们编写了图形程序之后再考虑。

当被问到“为什么讲授编程”这个问题时，计算机教育的先驱 John Kemeny 曾经把自顶向下的编程方法与我们熟悉的汽车来类比：你没必要知道汽车发动机罩下面是什么，但是如果知道内部的工作原理，就会去驾驶汽车而不是坐在汽车后座上。同样的类比适用于讲授计算机图形学的方法。第一种方法(算法的方法)是把汽车工作原理的方方面面都交代清楚：发动机、传动装置、

燃烧过程等。第二种方法(概览的方法)是雇一名司机,自己坐在后面观光。第三种方法(本书采用的编程方法)是教会你如何驾驶并把汽车开到你想要去的地方,正如一句过时的租赁广告词说的那样:“让我们把你放在驾驶员的位子上去。”

使用 WebGL 和 JavaScript 编程

当本书的第一作者 Edward Angel 在 30 年前开始讲授计算机图形学的时候,当时以面向编程的方式讲授这门课程并编写相应教材的最大障碍是缺乏一个广泛接受的图形库或者应用程序编程接口(API)。遇到的困难包括:高成本、可用性受限、缺乏通用性以及高复杂性。OpenGL 的开发解决了我们中很多人在其他 API 中遇到的以及在使用自己编写的软件作为替代时所面临的大多数困难,今天所有的平台都支持 OpenGL,它已被普遍认为是跨平台的标准。

一门图形学的课程远远不能仅讲授如何使用一种特定的 API,一种好的 API 应使得讲授图形学中的一些关键内容变得更容易,这些内容包括三维图形学、光照和着色、客户/服务器图形系统、建模以及实现算法。我们相信,OpenGL 的扩展功能以及良好定义的结构可以为讲授图形学理论和实践这两方面都提供更好的基础,而且 OpenGL 同样也可以为讲授包括纹理映射、图像融合以及可编程着色器在内的一些高级概念提供更好的基础。

大约 18 年前,Edward Angel 就开始在授课时使用 OpenGL,结果令他欣喜若狂。在那个学期的期中之前,所有的学生就都能编写出中等复杂程度的三维图形程序,而编写这样的程序需要理解三维显示和事件驱动输入。在这之前几年的计算机图形学教学中,Edward Angel 从未收到过如此好的教学效果,那次课程孕育出了本书的第一版。

本书是关于计算机图形学的教科书,并不是 OpenGL 或者 WebGL 的使用手册。因此,没有把 WebGL API 的方方面面都一一介绍,而是仅仅解释了对于掌握本书内容来说必要的那些部分。因而,本书在介绍 WebGL 时的定位是允许其他 API 用户毫无困难地阅读本书内容。

与前几个版本不同,这一版使用 WebGL 和 JavaScript 来编写所有的例子。WebGL 是使用 JavaScript 来实现 OpenGL ES 2.0,并且可以在当前大多数浏览器中运行。因为 HTML5 的支持,WebGL 不但提供了和其他应用程序的兼容性,而且不依赖于某个平台。WebGL 在浏览器中运行,并且充分利用主机的图形硬件。虽然 JavaScript 不是我们讲授编程课的常用语言,但它是一种 Web 语言。在过去的几年中,JavaScript 已经逐渐变得越来越强大,经验告诉我们,那些熟悉 Java、C 或者 C++ 的学生使用 JavaScript 来编程应该没有什么困难。

包括 WebGL 在内的所有的现代 OpenGL 版本都要求每个应用程序提供两个着色器,这两个着色器用 OpenGL 着色器语言(GLSL)编写。GLSL 类似于 C 语言,但增加了向量和矩阵作为基本类型,以及一些 C++ 的特性,例如操作符重载。我们已经添加了 JavaScript 库 MV.js,不仅支持我们使用的图形函数,还支持 GLSL 中的类型和操作。

适用的读者

本书适用于计算机科学和工程专业的高年级本科生和一年级研究生,以及具有良好程序设计能力的其他专业学生。本书对于许多专业人员也是有价值的。我们已经成功地为专业人员讲授了 100 多次短期课程,从这些非传统意义的学生身上获得的经验极大地影响了本书的内容。

在阅读本书之前,读者应该熟练掌握 JavaScript、C、C++ 语言或者 Java,理解链表和树等基本数据结构并具有线性代数和三角学的初步知识。我们发现计算机专业学生的数学背景相差很大,不管是本科生还是研究生都是如此。因此,本书包含了很多计算机基础图形学所需的线性代数和几何学知识。

本书的内容组织

本书内容组织如下：第1章概览整个图形学领域并介绍了基于光学设备的图像生成，这样我们从一开始就引入了三维图形学的概念。第2章介绍 WebGL 编程，虽然编写的第一个示例程序是二维的（每一章都有一个或者多个完整的编程示例），但是这个程序是嵌入在三维环境下编写的，并且可以扩展到三维。第3章介绍交互式图形学，并在浏览器环境中进行基于事件驱动的图形程序开发。第4章和第5章讨论三维图形学概念，第4章讨论如何定义和处理三维对象，而第5章讨论如何观察它们。第6章介绍光线和材质之间的相互作用以及着色。第7章介绍一些图形硬件和 WebGL 支持的新的离散技术。所有这些技术涉及了各种缓存。这些章节应该按顺序讲授，在一个总共 15 周的学期里，大约 10 周可以讲完这些内容。

后面的 5 章可以按照任何顺序阅读。所有这 5 章都或多或少是开放式的，读者可以概览，也可以深究其中的一些课题。第8章讨论实现，对三维图形绘制流水线中的每个基本步骤都给出了一种或两种主要算法，这些基本步骤包括裁剪、直线段生成和多边形填充。第9章包含层级建模主题中的若干松散内容，涉及从层级模型（封装了各部分之间联系的模型）的构造到互联网图形绘制的高级方法。这一章还包括了对场景图的介绍。第10章介绍若干过程方法，包括粒子系统、分形和过程噪声。第11章讨论曲线和曲面，包括细分曲面。第12章介绍其他一些高绘制方法，进一步讨论了光线跟踪和辐射度方法，还介绍了基于图像的绘制和并行绘制。

附录 A 提供了阅读、编译和链接应用程序和着色器所需 WebGL 函数的细节。附录 B 和附录 C 包含对背景数学知识的回顾，附录 D 讨论了利用奈奎斯特定理进行采样和反走样，并把结果应用到计算机图形学中。

相对第六版的变化

读者对于本书前六版的绝大多数评价都是肯定的，特别是对于采用 OpenGL 和自顶向下的方法更是如此。在第六版中，我们抛弃了固定功能渲染管线，而完全采用基于着色器的 OpenGL。在这一版中，我们使用 WebGL，WebGL 不仅是完全基于着色器的（每个程序必须至少提供一个顶点着色器和一个片段着色器），而且能运行于最新的 Web 浏览器。

所有的程序使用 JavaScript 编写。虽然 JavaScript 具有其自身的特性，但我们不希望具有高级语言如 Java、C 或者 C++ 使用经验的学生在使用它时遇到严重的困难。

正像我们在本版前言开始所指出的那样，每一个应用程序都必须提供着色器。因此，可编程着色器及 GLSL 需要在第2章介绍。与以前的版本相比，虽然第七版中的许多示例程序的输出结果与以前的版本相同，但它们的代码完全不同。

第六版不再将输入和交互作为单独一章，而是将有关内容合并到了其他章节。而在本版本中，我们对这部分内容又恢复为独立的章节。因为通过 HTML5 把事件驱动输入集成到 WebGL 是很方便灵活的，所以我们做出了这样的决定。

我们在本书中加入了有关离屏绘制和绘制到纹理的相关内容。这些技术已经成为一些利用 GPU 进行计算密集型应用程序开发的基础，例如图像处理和模拟。

读者对以前版本中第1章到第6章的核心内容反响很好，所以本版尽量保持这些章节的内容，使其改变降到最低。本版仍将第1章至第7章视为计算机图形学导论课程的核心内容，第8章至第12章可以按任何顺序来使用，既可以作为一个学期的概论课，也可以作为顺序开设的两个学期的课程内容。

辅助材料

在作者的网站 www.cs.unm.edu/~angel 上可以找到本书的辅助材料。所有读者都可以获得下面的辅助材料^①：

- 关于 WebGL 的资源信息
- 程序代码
- 部分习题的答案
- 讲稿课件
- 书中的插图

还有一些辅助材料，包括所有非程序设计习题的解答，仅提供给采用本书做教材的教师使用。请与贵校的 Pearson Education 代表联系或者访问网站 www.pearsonhighered.com/irc 以获取使用这些材料的信息^②。

致谢

在过去的几年中，Edward Angel 有幸在新墨西哥大学遇到了许多优秀的学生，是他们最早使他对 OpenGL 产生了兴趣，从他们那里 Edward Angel 学到了很多。这些学生包括：Ye Cong, Pat Crossno, Tommie Daniel, Chris Davis, Lisa Desjarlais, Kim Edlund, Lee Ann Fisk, Maria Gallegos, Brian Jones, Christopher Jordan, Takeshi Hakamata, Max Hazelrigg, Sheryl Hurley, Thomas Keller, Ge Li, Pat McCormick, Al McPherson, Ken Moreland, Martin Muller, David Munich, Jim Pinkerton, Jim Prewett, Dave Rogers, Hal Smyer, Dave Vick, Hue (Bumgarner-Kirby) Walker, Brian Wylie 和 Jin Xiong，许多彩图中的例子都出自这些学生。

本书第一版是 Edward Angel 在休假期间写作的，他在 5 个不同的国家完成了本书。如果没有若干人士以及若干大学、研究机构给他提供的帮助，该书是完不成的，他们给他提供了便利。衷心感谢委内瑞拉安第斯大学 (Universidad de los Andes) 的 Jonas Montilva 和 Chris Birkbeck、厄瓜多尔天球赤道理工大学 (Universidad Tecnológica Equinoccial) 的 Rodrigo Gallegos 和 Aristides Novoa、新竹清华大学的 Long Wen Chang、香港中文大学的 Kim Hong Wong 和 Pheng Ann Heng。Edward Angel 到许多地方的参观访问都得益于国际超导技术中心 (ISTEC) 和新墨西哥大学的 Ramiro Jordan 的帮忙。无论 Edward Angel 何时有问题，新墨西哥大学的 John Brayer 和 Jason Stewart、Addison-Wesley 出版社的 Helen Goldstein 都设法帮他解决。Edward Angel 的网站上描述了写作第一版的经历。

NVIDIA 公司的 David Kirk 和 Mark Kilgard 慷慨地为测试很多算法提供了显卡。还应感谢许多给予了极大帮助的人士，他们是 Ben Bederson, Gonzalo Cartagenova, Tom Caudell, Kathi Collins, Kathleen Danielson, Roger Ehrich, Robert Geist, Chuck Hansen, Mark Henne, Bernard Moret, Dick Nordhaus, Helena Saona, Dave Shreiner, Vicki Shreiner, Gwen Sylvan 和 Mason Woo。承蒙 OpenGL 协会的 Mark Kilgard、Brian Paul 和 Nate Robins 的极大帮助，他们编写了能使 OpenGL 代码在各种平台上都可以开发的软件。

新墨西哥大学的艺术、研究、技术和科学实验室 (ARTS Lab) 以及高性能计算中心对 Edward

① 部分资源也可登录华信教育资源网 (www.hxedu.com.cn) 免费注册下载——编者注。

② 为获取本书供教师使用的教学支持文档，可参阅本书后所附的“教学支持说明”——编者注。

Angel 的许多项目提供了支持, 计算机科学系、美术学院的艺术技术中心、国家科学基金、美国 Sandia 国家实验室和 Los Alamos 国家实验室资助了许多 Edward Angel 的学生和研究项目, 本书的部分内容就来自这些研究项目中的一部分。曾在北极星天文馆, 现在 ARTS 实验室的 David Beinling 为 FullDome 项目提供了极大的支持, Sheryl Hurley, Christopher Jordan, Laurel Ladwig, Jon Strawn 和 Hue (Bumgarner-Kirby) Walker 通过 FullDome 项目提供了本书彩图中的一些图像。Hue Walker 为前三个版本设计了精美的封面, 其画面是本版封面的基础。

Edward Angel 还要感谢 Santa Fe Complex 创立的非正式小组, 包括 Jeff Bowles, Ruth Chabay, Stephen Guerin, Bruce Sherwood, Scott Wittenberg, 尤其是推广 JavaScript 的 Owen Densmore, 他说服自己在 Santa Fe 讲授计算机图形学课程以深入研究 JavaScript, 使我们都获得了相关经验。

本书的第二作者 Dave Shreiner 首先感谢 Edward Angel 邀他参与了该书的写作, 我们彼此交换自己对 OpenGL 的看法以及如何开展教学已多年, 把那些概念教给新学生是很令人兴奋的。Dave Shreiner 也首先感谢曾在硅谷图形计算机系统工作过的同事们、创建 OpenGL 的同事们, 以及一直研究 API 的 Khronos 组织的 OpenGL 工作组。就像 Edward Angel 所说, ACM SIGGRAPH 显然在开发这些材料中起到了重要作用, 对于该组织能为探索我们的想法所提供的热点研究题目不胜感激。

评阅人在本书应该包括哪些内容和应该介绍到什么程度这两方面提出了各种各样的看法, 先前版本的评阅人包括: Gur Saran Adhar (北卡罗来纳州立大学威明顿分校), Mario Agrular (杰克逊维尔州立大学), Michael Anderson (哈特福德大学), Norman I. Badler [宾夕法尼亚大学 (University of Pennsylvania)], Mike Bailey (俄勒冈州立大学), Marty Barrett (东田纳西州立大学), C. S. Bauer (中佛罗里达大学), Bedrich Benes (普度大学), Kabekode V. Bhat [宾夕法尼亚州立大学 (The Pennsylvania State University)], Isabelle Bichindartiz (华盛顿大学), Cory D. Boatright (宾夕法尼亚大学), Eric Brown 和 Robert P. Burton (杨伯翰大学), Sam Buss (加利福尼亚州立大学圣迭戈分校), Kai H. Chang (奥本大学), James Cremer (艾奥瓦大学), Ron DiNapoli (康奈尔大学), John David N. Dionisio (洛约拉马利蒙特大学), Eric Alan Durant (密尔瓦基工程学校), David S. Ebert (普度大学), Richard R. Eckert (宾厄姆顿大学), W. Randolph Franklin (伦斯勒理工学院), Natacha Gueorguieva (纽约城市大学/斯塔腾岛学院), Jianchao (Jack) Han (加利福尼亚州立大学多明戈兹分校), Chenyi Hu (阿肯色中央大学), George Kamberov (史蒂文斯理工学院), Mark Kilgard (NVIDIA 公司), Lisa B. Lancor (南康涅狄格州立大学), Chung Lee (加州波莫纳理工大学), John L. Lowther (密歇根理工大学), R. Marshall (波士顿大学和布里奇沃特州立学院), Hugh C. Masterman (马萨诸塞大学洛威尔分校), Bruce A. Maxwell (史威特摩尔学院), Tim McGraw (西弗吉尼亚大学), James R. Miller (堪萨斯州立大学), Rodrigo Obando (哥伦布州立大学), Jon A. Preston (南方州立理工大学), Andrea Salgian (新泽西学院), Lori L. Scarlatos (纽约城市大学布鲁克林学院), Han-Wei Shen (俄亥俄州立大学), Oliver Staadt (加利福尼亚州立大学戴维斯分校), Stephen L. Stepoway (南卫理公会大学), Bill Toll (泰勒大学), Michael Wainer (南伊利诺伊州立大学卡本戴尔分校), YangWang (南卫理公会州立大学), Steve Warren (堪萨斯州立大学), Mike Way (佛罗里达南方学院), George Wolberg (纽约城市大学), Xiaoyu Zhang (加利福尼亚州立大学圣马科斯分校), Ye Zhao (肯特州立大学) 和 Ying Zhu (佐治亚州立大学), 虽然最后的定稿可能没有反映出他们的意见 (因为有些意见彼此相左), 但每位评阅人的意见都鞭策着我们反思书稿的每一页内容。

本版评阅人给予了极大的帮助和支持, 他们是 Mike Bailey (俄勒冈州立大学), Patrick Cozzi (宾夕法尼亚大学和分析图形公司) 和 Jeff Parker (哈佛大学)。他们都对以前的版本很熟悉, 并且很愿意转向本书的 WebGL 编程。

我们同样感谢 Addison-Wesley 出版社的整个出版团队。在本书的七个版本和 *OpenGL Primer* (OpenGL 程序设计指南) 的出版过程中, Edward Angel 与 Peter Gordon、Maite Suarez-Rivas 和 Matt Goldstein 三位编辑合作得非常愉快。对于本书, Marilyn Lloyd 和 Kayla Smith-Tarbox 亲自提供了大量帮助。在本书全部七个版本出版时, Paul Anagnostopoulos 一直帮助解决使用 Windfall Software 工具时的排版问题, 帮助之大溢于言表。Edward Angel 要特别感谢编辑 Lyn Dupré, 如果读者看到第一版的原稿, 就会惊叹于 Lyn Dupré 的妙笔神功。

Edward Angel 在此要特别鸣谢妻子 Rose Mary Molnar, 她为本书第一版制作了插图, 其中许多插图构成了本书插图的基础。也许只有经历过图书出版过程并撰写过许多文稿的人以及为出版工作做出过贡献的人才能充分理解书写过程的艰辛。真诚地把本书献给 Rose Mary, 尽管这仍不足以表达 Edward Angel 对妻子所有贡献的谢意。

Dave Shreiner 感激妻子 Vicki 的支持和鼓励, 没有她, 本书创造性的成果不会展现给读者; 她不仅给予 Dave Shreiner 温暖, 陪伴左右, 而且对本书的描述和素材提出了珍贵意见。在 Dave Shreiner 对 OpenGL 的所有研究中, 她都称得上是一个不可多得的幕后搭档。

Edward Angel
Dave Shreiner

目 录

第1章 图形系统和模型	1	建议阅读资料	27
1.1 计算机图形学的应用	1	习题	27
1.1.1 信息的显示	1	第2章 图形学编程	29
1.1.2 设计	2	2.1 Sierpinski 镂垫	29
1.1.3 仿真和动画	2	2.2 编写二维图形应用程序	31
1.1.4 用户界面	3	2.3 WebGL 应用程序编程接口	35
1.2 图形系统	3	2.3.1 图形函数	35
1.2.1 像素和帧缓存	4	2.3.2 图形绘制流水线和状态机	36
1.2.2 CPU 与 GPU	5	2.3.3 OpenGL 和 WebGL	37
1.2.3 输出设备	5	2.3.4 WebGL 接口	37
1.2.4 输入设备	7	2.3.5 坐标系	38
1.3 物理图像与合成图像	7	2.4 图元和属性	39
1.3.1 对象和观察者	8	2.4.1 多边形的基本概念	41
1.3.2 光和图像	9	2.4.2 WebGL 中的多边形	42
1.3.3 成像模型	10	2.4.3 利用多边形近似球面	43
1.4 成像系统	11	2.4.4 三角剖分	43
1.4.1 针孔照相机	11	2.4.5 文本	44
1.4.2 人类视觉系统	12	2.4.6 曲线和曲面对象	45
1.5 虚拟照相机模型	13	2.4.7 属性	45
1.6 应用程序编程接口	15	2.5 颜色	46
1.6.1 笔式绘图仪模型	15	2.5.1 RGB 颜色	47
1.6.2 三维 API	17	2.5.2 索引颜色	49
1.6.3 使用 WebGL API 绘制的一系列 图像	19	2.5.3 设置颜色属性	50
1.6.4 建模-绘制模式	20	2.6 观察	51
1.7 图形绘制系统的体系结构	21	2.6.1 正投影	51
1.7.1 显示处理器	21	2.6.2 二维观察	52
1.7.2 流水线体系结构	21	2.7 控制函数	53
1.7.3 图形绘制流水线	22	2.7.1 和窗口系统的交互	53
1.7.4 顶点处理	22	2.7.2 宽高比和视口	54
1.7.5 裁剪和图元组装	23	2.7.3 应用程序组织结构	55
1.7.6 光栅化	23	2.8 Sierpinski 镂垫绘制程序	56
1.7.7 片元处理	23	2.8.1 向 GPU 发送数据	58
1.8 可编程流水线	24	2.8.2 绘制 Sierpinski 镂垫的 点数据	59
1.9 性能特征	24	2.8.3 顶点着色器	59
1.10 OpenGL 版本和 WebGL	25	2.8.4 片元着色器	60
小结和注释	26	2.8.5 组合代码段	60

2.8.6	initShaders()函数	61	3.10	建立交互式模型	95
2.8.7	init()函数	62	3.11	交互式程序的设计	99
2.8.8	从应用程序读入着色器	63		小结和注释	99
2.9	多边形和递归	63		建议阅读资料	100
2.10	三维 Sierpinski 镂垫	65		习题	101
2.10.1	使用三维点	65	第4章	几何对象和变换	103
2.10.2	命名规范	66	4.1	标量、点和向量	103
2.10.3	使用多边形的三维 Sierpinski 镂垫	66	4.1.1	几何对象	103
2.10.4	隐藏面消除	69	4.1.2	与坐标无关的几何	105
	小结和注释	70	4.1.3	数学的观点: 向量空间和仿射空间	106
	建议阅读资料	71	4.1.4	计算机科学的观点	106
	习题	72	4.1.5	几何 ADT	107
第3章	交互和动画	75	4.1.6	直线	108
3.1	动画	75	4.1.7	仿射加法	108
3.1.1	旋转的正方形	75	4.1.8	凸性	108
3.1.2	显示过程	77	4.1.9	点积和叉积	109
3.1.3	双缓存	78	4.1.10	平面	110
3.1.4	使用定时器	79	4.2	三维图元	111
3.1.5	使用 requestAnimationFrame 函数	79	4.3	坐标系和标架	112
3.2	交互	80	4.3.1	向量的表示和 n 元组	114
3.3	输入设备	81	4.3.2	坐标系的变换	114
3.4	物理输入设备	82	4.3.3	举例: 不同基下的表示之间的变换	116
3.4.1	键盘码	82	4.3.4	齐次坐标	117
3.4.2	光笔	82	4.3.5	举例: 标架变换	119
3.4.3	鼠标和跟踪球	82	4.3.6	对表示进行处理	120
3.4.4	数据板、触摸板和触摸屏	83	4.4	WebGL 中的标架	122
3.4.5	操作杆	84	4.5	矩阵和向量类型	125
3.4.6	多维输入设备	84	4.6	建模一个彩色立方体	127
3.4.7	逻辑设备	85	4.6.1	建模立方体的面	127
3.4.8	输入模式	86	4.6.2	向内和向外的面	128
3.5	客户和服务器	87	4.6.3	表示对象的数据结构	128
3.6	事件驱动输入编程	88	4.6.4	彩色立方体	129
3.6.1	事件和事件监听器	88	4.6.5	颜色插值	130
3.6.2	增加按钮	89	4.6.6	显示立方体	131
3.6.3	菜单	90	4.6.7	使用元素绘制网格	132
3.6.4	使用键码	91	4.7	仿射变换	133
3.6.5	滑动条	91	4.8	平移、旋转和缩放	135
3.7	位置输入	92	4.8.1	平移	135
3.8	窗口事件	94	4.8.2	旋转	135
3.9	拾取	94	4.8.3	缩放	137

4.9 变换的齐次坐标表示	137	5.3.4 其他观察 API	177
4.9.1 平移	138	5.4 平行投影	178
4.9.2 缩放	138	5.4.1 正投影	178
4.9.3 旋转	139	5.4.2 WebGL 中的平行投影	179
4.9.4 错切	140	5.4.3 投影的规范化	179
4.10 变换的级联	141	5.4.4 正投影变换矩阵	180
4.10.1 不动点在任意位置的 旋转	142	5.4.5 斜投影	181
4.10.2 一般的旋转	142	5.4.6 交互式观察立方体	184
4.10.3 实例变换	143	5.5 透视投影	185
4.10.4 绕任意轴的旋转	144	5.6 WebGL 中的透视投影	188
4.11 WebGL 中的变换矩阵	147	5.7 透视投影变换矩阵	189
4.11.1 当前变换矩阵	147	5.7.1 透视投影的规范化	190
4.11.2 基本矩阵函数	148	5.7.2 WebGL 中的透视投影 变换	192
4.11.3 旋转、平移和缩放	148	5.7.3 透视投影示例程序	193
4.11.4 绕任意不动点的旋转	149	5.8 隐藏面消除	193
4.11.5 变换的顺序	149	5.9 显示网格	195
4.12 使立方体旋转起来	150	5.9.1 显示网格曲面	198
4.13 三维应用程序的接口	153	5.9.2 多边形偏移	199
4.13.1 使用屏幕区域	154	5.9.3 在场景中漫游	200
4.13.2 虚拟跟踪球	154	5.10 投影和阴影	200
4.13.3 平滑的旋转	155	5.11 阴影映射	204
4.13.4 增量式旋转	156	小结和注释	204
4.14 四元数	157	建议阅读资料	205
4.14.1 复数和四元数	157	习题	205
4.14.2 四元数和旋转	158	第 6 章 光照和着色	207
4.14.3 四元数和万向节死锁	159	6.1 光线和材质	207
小结和注释	160	6.2 光源	209
建议阅读资料	160	6.2.1 彩色光源	210
习题	161	6.2.2 环境光	210
第 5 章 观察	164	6.2.3 点光源	211
5.1 经典观察和计算机观察	164	6.2.4 聚光灯	212
5.1.1 经典观察	165	6.2.5 远距离光源	212
5.1.2 正投影	166	6.3 Phong 反射模型	213
5.1.3 轴测投影	166	6.3.1 环境光反射	214
5.1.4 斜投影	167	6.3.2 漫反射	215
5.1.5 透视观察	168	6.3.3 镜面反射	216
5.2 计算机观察	169	6.3.4 改进的 Phong 模型	217
5.3 定位照相机	170	6.4 计算向量	218
5.3.1 照相机标架的定位	170	6.4.1 法向量	218
5.3.2 两个观察 API	174	6.4.2 反射角	220
5.3.3 lookAt 函数	176	6.5 多边形的着色	221

6.5.1 均匀着色	221	7.10.4 再探反走样	280
6.5.2 平滑着色	222	7.10.5 从后向前或从前向后绘制 多边形	282
6.5.3 Phong 着色	223	7.10.6 场景反走样和多重采样	282
6.6 通过递归细分逼近球面	224	7.10.7 图像处理	283
6.7 指定光照参数	226	7.10.8 其他的多通道绘制方法	284
6.7.1 光源	226	7.11 GPGPU	284
6.7.2 材质	227	7.12 帧缓存对象	287
6.8 实现光照模型	227	7.13 缓存交换	292
6.8.1 在 WebGL 应用程序代码中应用 光照模型	228	7.14 拾取	294
6.8.2 绘制效率	230	小结和注释	298
6.8.3 在顶点着色器中实现光照 计算	231	建议阅读资料	299
6.9 球面模型的着色	234	习题	299
6.10 基于每个片元的光照计算	235	第 8 章 从几何到像素	301
6.11 非真实感着色	237	8.1 图形绘制流水线的基本实现 策略	301
6.12 全局光照	238	8.2 图形绘制系统的四个主要 任务	303
小结和注释	239	8.2.1 建模	303
建议阅读资料	239	8.2.2 几何处理	304
习题	240	8.2.3 光栅化	305
第 7 章 离散技术	242	8.2.4 片元处理	305
7.1 缓存	242	8.3 裁剪	306
7.2 数字图像	244	8.4 线段的裁剪	306
7.3 映射方法	246	8.4.1 Cohen-Sutherland 裁剪 算法	307
7.4 二维纹理映射	248	8.4.2 Liang-Barsky 裁剪算法	308
7.5 WebGL 的纹理映射	252	8.5 多边形的裁剪	310
7.5.1 纹理对象	253	8.6 其他图元的裁剪	312
7.5.2 纹理图像数组	253	8.6.1 包围盒与包围体	312
7.5.3 纹理坐标和纹理采样器	255	8.6.2 曲线、曲面和字符的裁剪	313
7.5.4 纹理采样	259	8.6.3 帧缓存中的裁剪	314
7.5.5 使用纹理坐标	261	8.7 三维裁剪	314
7.5.6 多纹理映射	262	8.8 光栅化	316
7.6 纹理生成	264	8.9 Bresenham 算法	318
7.7 环境贴图	265	8.10 多边形光栅化算法	320
7.8 反射贴图示例程序	268	8.10.1 内外测试法	320
7.9 凹凸映射	271	8.10.2 WebGL 与凹多边形	321
7.9.1 计算凹凸贴图	272	8.10.3 填充与分类	322
7.9.2 凹凸贴图示例程序	274	8.10.4 漫水填充算法	322
7.10 融合技术	278	8.10.5 处理奇点	323
7.10.1 不透明度与融合	278	8.11 隐藏面消除	323
7.10.2 图像融合	279		
7.10.3 WebGL 中图像的融合	279		

8.11.1 对象空间和图像空间消隐 算法	324	10.2 基于物理的建模和粒子系统 ...	370
8.11.2 排序与隐藏面消除	325	10.3 牛顿粒子	371
8.11.3 扫描线填充算法	325	10.3.1 独立的粒子	373
8.11.4 背面剔除	326	10.3.2 弹簧力	373
8.11.5 z 缓存算法	327	10.3.3 吸引力与排斥力	374
8.11.6 基于 z 缓存的扫描转换 算法	329	10.4 求解粒子系统方程	375
8.11.7 深度排序和画家算法	330	10.5 约束条件	377
8.12 反走样	332	10.5.1 碰撞	377
8.13 显示方面的问题	333	10.5.2 软约束	379
8.13.1 颜色系统	334	10.6 一个简单的粒子系统	379
8.13.2 颜色矩阵	336	10.6.1 绘制粒子	380
8.13.3 γ 校正	337	10.6.2 更新粒子的位置	380
8.13.4 抖动输出技术和半色调输出 技术	337	10.6.3 碰撞	380
小结和注释	338	10.6.4 作用力	381
建议阅读资料	339	10.6.5 群聚行为	382
习题	340	10.7 基于 agent 的建模	383
第 9 章 层级建模方法	342	10.8 基于语言的建模	384
9.1 图符和实例	342	10.9 递归方法和分形	387
9.2 层级模型	344	10.9.1 标尺和长度	388
9.3 机器人手臂	345	10.9.2 分形维数	388
9.4 树与遍历	348	10.9.3 中点划分及布朗运动	389
9.5 使用树数据结构	351	10.9.4 分形山脉	390
9.6 动画	354	10.9.5 Mandelbrot 集	391
9.7 图形对象	355	10.9.6 Mandelbrot 片元着色器 ...	393
9.7.1 方法、属性和消息	356	10.10 过程噪声	394
9.7.2 一个立方体对象	357	小结和注释	396
9.7.3 对象与层级结构	357	建议阅读资料	397
9.7.4 几何和非几何对象	358	习题	397
9.8 场景图	359	第 11 章 曲线和曲面	399
9.9 实现场景图	361	11.1 曲线和曲面的表示形式	399
9.10 其他树结构	362	11.1.1 显式表示形式	399
9.10.1 CSG 树	362	11.1.2 隐式表示形式	400
9.10.2 BSP 树	363	11.1.3 参数表示形式	401
9.10.3 四叉树和八叉树	365	11.1.4 参数多项式曲线	402
小结和注释	367	11.1.5 参数多项式曲面	402
建议阅读资料	367	11.2 设计准则	403
习题	368	11.3 三次参数多项式曲线	404
第 10 章 过程建模方法	369	11.4 插值	405
10.1 基于算法的建模	369	11.4.1 调和函数	406
		11.4.2 三次插值曲面片	407
		11.5 Hermite 曲线和曲面	408
		11.5.1 Hermite 形式	409

11.5.2	几何与参数连续性	410	12.3.2	计算交点	443
11.6	Bézier 曲线和曲面	411	12.3.3	其他不同形式的光线跟踪	445
11.6.1	Bézier 曲线	411	12.4	绘制方程	446
11.6.2	Bézier 曲面片	413	12.5	辐射度方法	448
11.7	三次 B 样条	413	12.5.1	辐射度方程	449
11.7.1	三次 B 样条曲线	414	12.5.2	求解辐射度方程	450
11.7.2	B 样条和基函数	416	12.5.3	计算排列因子	450
11.7.3	样条曲面	416	12.5.4	实现辐射度算法	452
11.8	普通 B 样条	417	12.6	全局光照和路径跟踪	453
11.8.1	B 样条的递归定义	417	12.7	RenderMan	454
11.8.2	均匀样条	418	12.8	并行绘制	455
11.8.3	非均匀 B 样条	418	12.8.1	sort-middle 绘制方法	456
11.8.4	NURBS	419	12.8.2	sort-last 绘制方法	457
11.8.5	Catmull-Rom 样条	420	12.8.3	sort-first 绘制方法	459
11.9	曲线和曲面的绘制	420	12.9	GPU 硬件实现	460
11.9.1	多项式求值方法	421	12.10	隐函数和等高线图	461
11.9.2	递归细分 Bézier 多项式	422	12.10.1	步进方格	461
11.9.3	基于细分算法的其他多项式曲线的绘制	423	12.10.2	步进三角形	465
11.9.4	细分 Bézier 曲面	424	12.11	体绘制	466
11.10	Utah 茶壶模型	425	12.11.1	体数据集	466
11.11	代数曲面	427	12.11.2	隐函数的可视化	467
11.11.1	二次曲面	427	12.12	等值面与步进立方体	468
11.11.2	使用光线跟踪绘制曲面	428	12.13	步进四面体	470
11.12	曲线和曲面的细分	429	12.14	网格简化	471
11.13	从数据生成网格	431	12.15	直接体绘制	471
11.13.1	回顾高度场	431	12.15.1	指定颜色和透明度	472
11.13.2	Delaunay 三角剖分	432	12.15.2	抛雪球算法	472
11.13.3	点云	435	12.15.3	体光线跟踪	473
11.14	支持曲线和曲面的图形 API	435	12.15.4	基于纹理映射的体绘制	474
11.14.1	曲面细分着色	435	12.16	基于图像的绘制	474
11.14.2	几何着色	436	小结和注释	476	
小结和注释		436	建议阅读资料	477	
建议阅读资料		437	习题	478	
习题		437	附录 A	初始化着色器	480
第 12 章	高级绘制	439	附录 B	空间	484
12.1	超越流水线绘制结构	439	附录 C	矩阵	490
12.2	光线跟踪	440	附录 D	采样与走样	497
12.3	构建一个简单的光线跟踪器	442	参考文献		503
12.3.1	光线跟踪递归算法	442			