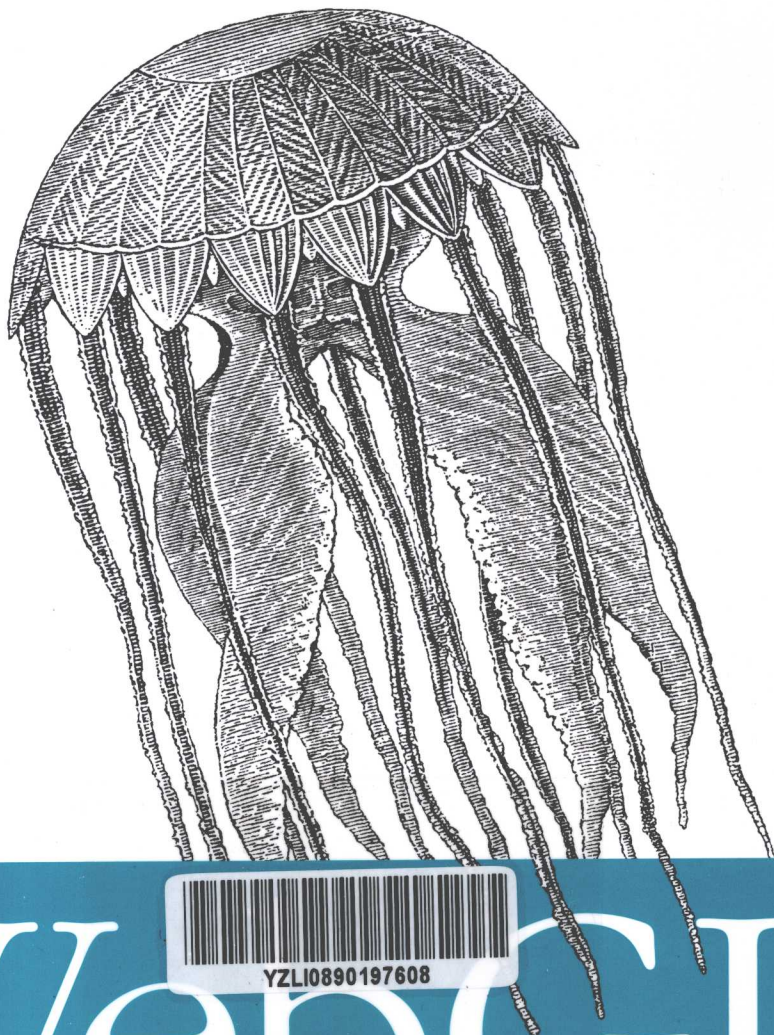


WebGL: Up and Running



WebGL

入门指南

[美] Tony Parisi 著
郝稼力 译

O'REILLY®

 人民邮电出版社
POSTS & TELECOM PRESS

WebGL 入门指南



[美] *Tony Parisi* 著

郝稼力 译



YZLI0890197608

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

O'Reilly Media, Inc. 授权人民邮电出版社出版

人 民 邮 电 出 版 社
北 京

图书在版编目 (C I P) 数据

WebGL入门指南 / (美) 帕里西 (Parisi, T.) 著 ;
郝稼力译. — 北京 : 人民邮电出版社, 2013. 6
ISBN 978-7-115-31668-4

I. ①W… II. ①帕… ②郝… III. ①网页制作工具
IV. ①TP393. 092

中国版本图书馆CIP数据核字(2013)第077233号

版权声明

Copyright ©2012 by O'Reilly Media, Inc.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and Posts & Telecom Press, 2013. Authorized translation of the English edition, 2013 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

本书中文简体版由 **O'Reilly Media, Inc.** 授权人民邮电出版社出版。未经出版者书面许可, 对本书的任何部分不得以任何方式复制或抄袭。

版权所有, 侵权必究。

-
- ◆ 著 [美] Tony Parisi
 - 译 郝稼力
 - 责任编辑 杨海玲
 - 责任印制 程彦红 焦志炜
 - ◆ 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 电子邮件 315@ptpress.com.cn
 - 网址 <http://www.ptpress.com.cn>
 - 三河市海波印务有限公司印刷
 - ◆ 开本: 787×1000 1/16
 - 印张: 12.75
 - 字数: 254 千字 2013 年 6 月第 1 版
 - 印数: 1 - 3 000 册 2013 年 6 月河北第 1 次印刷
 - 著作权合同登记号 图字: 01-2012-7987 号
-

定价: 49.00 元

读者服务热线: (010)67132692 印装质量热线: (010)67129223
反盗版热线: (010)67171154

内容提要

WebGL 是一项新的 Web 3D 图形标准，也是 HTML5 大家庭中的一员。本书从 WebGL 和 3D 图形学的基础概念讲起，循序渐进，用多个简单的实例直观地讲解了各个知识点，包括从坐标系统到投影矩阵这些数学基础，也包括从纹理贴图到模型动画这些图形效果；随后又理论结合实际，介绍了在现实开发环境中需要注意的各种问题，例如，如何挑选 WebGL 框架以及使用哪种 3D 内容交换格式；在最后一章，作者把全书所有讲过的知识综合到一起，制作了一个完整的 WebGL 游戏，让读者能够基本了解 WebGL 这一新技术的开发流程，以便读者可以独立开发自己的 WebGL 应用。

除此之外，在多个实例中作者还应用了一些非常便捷有效的图形开发技巧，这些脱胎于作者多年图形经验的小技巧可以节省很多工作量，也让初入图形学门槛的读者可以眼前一亮，领会到图形开发的美妙和乐趣。

本书适合 Web 开发人员阅读，尤其是对 3D 开发感兴趣的读者。

O'Reilly Media, Inc.介绍

O'Reilly Media通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自1978年开始，O'Reilly一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O'Reilly的发展充满了对创新的倡导、创造和发扬光大。

O'Reilly为软件开发人员带来革命性的“动物书”；创建第一个商业网站（GNN）；组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了Make杂志，从而成为DIY革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O'Reilly的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创新产业的革命性思想。作为技术人士获取信息的选择，O'Reilly现在还将先锋专家的知识传递给普通的计算机用户。无论是通过书籍出版，在线服务或者面授课程，每一项O'Reilly的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

业界评论

“O'Reilly Radar博客有口皆碑。”

——Wired

“O'Reilly凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。”

——Business 2.0

“O'Reilly Conference是聚集关键思想领袖的绝对典范。”

——CRN

“一本O'Reilly的书就代表一个有用、有前途、需要学习的主题。”

——Irish Times

“Tim是位特立独行的商人，他不光放眼于最长远、最广阔的视野并且切实地按照Yogi Berra的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去Tim似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——Linux Journal

译者序

自从互联网诞生以来，如何让网页变得更绚丽、更美观一直是人们在不断追求的目标。为了达到这个目的，我们发明了 HTML 标记语言来确定网页的结构，发明了 CSS 层叠式样式表来规范网页的外观，发明了 JavaScript 来实现各种特效，发明了 Flash 等插件来嵌入视频和动画效果。但是直到 WebGL 技术的出现，人们才得以从计算机图形学这一学术根基上解决了这一问题。由于直接和 GPU 交互，如果愿意的话，设计师可以用 WebGL 精确控制网页中的每一个像素。这是之前所有的技术都无法企及的高度。

WebGL 技术诞生的年代正是 Web 2.0 时代的后期，人们正在急切地寻找下一代互联网应当前进的方向。HTML5 被业界认为是合理的答案之一。使用了 Canvas 的 WebGL 正好贴合 HTML5，成为了这个大家庭中的一分子。与传统的前端技术以及 HTML5 中其他技术成员不同的是，WebGL 是被设计成直接工作在显卡（GPU）端的，所以它比目前其他的典型的 Web 技术都要更加复杂。总的来说，WebGL 的原生 API 是相当的低等级的。但这也正是它为什么能够如此迅速地大量的计算，并实现复杂的实时 3D 渲染的原因。另外，由于 WebGL 师承 OpenGL ES，而 OpenGL ES 又脱胎自 OpenGL，这一切都离不开计算机图形学，而这门科学对于之前所有的前端工程师来说，都是很少染指的。

于是 WebGL 的教学和普及陷入了一个两难境地，图形工程师不懂前端，而前端工程师更不懂图形。从这个角度上说，本书很好地解决了这个问题。作者用简单易懂的语言介绍了图形学的基本概念，同时又兼顾 Web 开发中的疑点和难点，无论读者之前专注于哪个领域，都能够很好地入门 WebGL 开发。另外，本书最后一章也是全书最大的亮点之一，在这一部分作者综合全书的知识点，一步一步陪着读者搭建了一个完整的 WebGL 游戏。为读者之后独立搭建自己的 WebGL 应用奠定了基础。这对于目前 WebGL 实验性示例频出但完整应用依然匮乏的局面来说，有显著的推动意义。

本书中文版翻译之时，发生了两件事。一是 Google Chrome 浏览器移除了 WebGL 上下文关键字中的“实验性”描述，二是微软 Internet Explorer 11 被发现开始支持 WebGL。前者标志着 WebGL 已经离开实验阶段，正式扬帆起航；后者或许意味着 WebGL 难得地作为一项前沿先锋技术即将实现所有浏览器厂商的大一统局面。也许未来，WebGL 会像今天的 CSS 一样普遍，真真切切地改变着互联网的外观，改变我们的生活！

译者简介

郝稼力 国内最大的 WebGL 技术社区及资讯站 HiWebGL.com 的创始人，首次把业界知名度最高的 WebGL 教程《LearningWebGL》翻译为中文，受到众多 WebGL 学习者的好评；致力于推动 WebGL 技术在国内的普及，曾在北京、上海等多地举办 WebGL 技术宣讲和讨论会。目前，正和伙伴走在创业的道路上，运营着国内首个 WebGL 网站——捞 3D 网站（Lao3D.com）。

序

1996 年夏天，我曾在 Silicon Graphic 公司的 Cosmo Software 部门做暑期实习生，这家公司正在开发用于 Web 浏览器中的 VRML（Virtual Reality Markup Language）播放器。VRML 第一次将交互式的 3D 图形带入万维网中。当时的 Web 还很稚嫩，在如此早期的时候，能在浏览器中看到整合的 3D 体验还是相当令人激动的。

遗憾的是，VRML 并没有获得原本想象中的那么多的支持和使用。从纯技术角度讲，有两个原因：首先，VRML 的可编程性受到性能孱弱的脚本语言虚拟机的限制，也就是说，人们不可能编写出用于整个 3D 场景的通用代码，这也就从根本上限制了 VRML 的应用领域；其次，VRML 的渲染模型是基于原始的 OpenGL API 的固化图形管线的。这意味着除了内置于系统中的视觉效果之外，人们无法添加新的自定义效果。

在随后的 16 年中，图形技术和编程语言获得了翻天覆地的变化。3D 图形管线也变成了完全可编程的，这意味着美术人员和设计师可以凭借他们的想象力创造出无穷的光影着色效果。另外，JavaScript 的虚拟机的性能也得到了巨大提升，流畅地改变 3D 场景的外观成为可能，甚至可以完全改变三角形的每个顶点。这样的灵活性第一次让在浏览器中自由地编写 3D 应用成为可能。

随着近几年来 OpenGL ES 2.0 API 的发展，计算机图形学的研发工作达到了顶峰；WebGL API 正是基于 OpenGL ES 2.0 API 的。后者是一个轻量级的纯基于着色器的图形库，广泛应用于目前几乎所有的新的智能手机和移动设备。WebGL 工作组和开发社区希望将 3D 图形处理器的威力以健壮和安全的方式释放到 Web 上，掀起新的激动人心的 3D Web 应用的浪潮，覆盖每一种操作系统和每一种计算设备。

Tony 的这本书通俗易懂，涵盖了在 Web 上开发 3D 应用的大部分实用知识。这本书既能帮助初学者快速入门，也包含了相当多的高级开发技巧，这部分信息即使对于资深的 3D 图形专家来说，也会眼前一亮。Tony 从显示基本的 3D 网格开始，不断介绍一些很有趣的内容，如视觉效果、动画、交互、创作 3D 内容等，这些内容组合在一起形成了一个 3D 游戏的原型。这本书非常好看，我非常喜欢，希望你也同样喜欢。

——Ken Russell

科纳斯组织（Khronos Group）WebGL 工作组主席

前言

早在 1994 年，Tim Berners-Lee^①就提出了一个倡议，呼吁创建一个用于网络虚拟现实的行业规范；Mark Pesce 和我做出了回应。因为当时我们只能负担得起一张飞机票，所以我们就派 Mark 去了日内瓦，并在全球第一次万维网开发者大会上展示我们的 Labyrinth 原型。在会上，Mark 对所有与会人员宣称我们已经完成了这样的一个规范。我们的演示非常简单，但是却直击要点——点击之后，在浏览器窗口中打开并点击超链接，然后显示一个可以旋转的 3D 物体。几周之内，我们创建了一个邮件列表，并拥有了虚拟现实标记语言（Virtual Reality Markup Language, VRML）——这一致力于将 3D 内容推向网络世界的开创性努力正式起航了。

15 年之后，3D 内容终于可以融入浏览器之中，尽管并不是采用 Mark 和我当年设想的方式。VRML 是一个很好的尝试，但是却过于超前——当时宽带连接和专用的图形硬件都还没有成为主流。VRML 启发了后来者和一些模仿者，但他们又都半途而废了。技术大潮起起伏伏，人们却未曾停止过对这一问题的钻研。最终，在 2009 年，我们终于迎来转折点：整个业界都被一个称为 WebGL 的新规范凝聚在一起。到现在，2012 年，看起来 WebGL 已经稳稳地扎根于整个行业。

WebGL 将 3D 内容带入浏览器之中，通过 JavaScript 与电脑中的图形硬件进行交互。基于 OpenGL ES（同样的图形内容可以运行于你的智能手机和平板电脑中）的特性，使得 WebGL 得到了发展并且得到了主流桌面和移动浏览器厂商的广泛支持。有了 WebGL，任何一个程序员都可以通过网络创造出震撼的图形效果，并传播给数以百万计的用户，例如无需下载的游戏、大数据的可视化、产品展示、模拟训练等众多应用。

尽管目前关于 WebGL 的网络资源已经非常丰富，而且许多框架和工具都开始支持 WebGL，但是 WebGL 的信息却依然散乱不堪，覆盖面非常有限。WebGL 的原生 API 是相当低等级的，对于大部分传统程序员来说都是不可及的，只有有经验的 3D 程序员才能驾驭它的原始形式。然而，原本 WebGL 的前提和愿景就是让 3D 内容可以走近每一个人，这其中既包括那些拥有当前电脑硬件配置并安装了当前浏览器的消费者，也包括任何一个有文本编辑器和产品设想的开发人员。我们需要找到一种方式，在 WebGL 的强大和其基础知识、使用工具之间搭起一座桥梁，使得人们可以练习编写 WebGL 应用。

^① Tim Berners-Lee 是万维网的发明者、互联网之父。1989 年 3 月他正式提出万维网的设想；1990 年 12 月 25 日，他在日内瓦的欧洲粒子物理实验室里开发出了世界上第一个网页浏览器。

这就是我写这本书的原因。

读者对象

如果你是一位 Web 程序员或设计师，这本书可以教会你使用 WebGL。本书假设你已经拥有了 HTML、CSS、JavaScript 的开发经验，并且熟悉 jQuery 和 Ajax。你不需要任何 3D 图形学经验，本书的第 1 章会简短介绍 3D 图形学的基础知识；此外，你不必是一位专家级的游戏开发者。

本书是如何组织的

本书共分三个部分，总计 8 章，另外还有一个附录。

- 第 1 章和第 2 章介绍了 WebGL API 和 Three.js，后者是一个开源的 JavaScript 库，本书的编程示例将使用此库。
- 第 3 章到第 6 章会深入介绍图形编程、动画和用户交互的细节，并会探索如何突破 WebGL 极限，把 2D 内容和 3D 内容无缝整合到用户体验中。
- 第 7 章和第 8 章将探讨如何在实际项目中编写 WebGL，从编程工具、文件格式到构建健壮安全的 WebGL 应用。在第 8 章中，我们将会构建第一个完整的应用，一个赛车游戏。
- 附录 A 列出了更多学习 WebGL 开发的资源，以及 WebGL 标准和相关的技术及工具。

大部分章节都遵循同一个套路：先解释概念，然后再深入代码当中举例说明，然后再返回来总结一下，接着再继续下一个话题。有时候我会花时间来写一些我内心的东西，或者对一些最佳实践或其他内容写一些豪言壮语。我希望我在传授知识和理论布道之间取得平衡，非常欢迎你联系我，并让我知道我做的到底怎么样。

如果你卡在了某个示例代码，那么请暂时略过并回头再看。你可以一直只读那些概念章节，然后在全部读完并做好准备之后，再开始看代码。另外，请不要犹豫，一定要在你最喜欢的支持 WebGL 的浏览器中打开这些示例，并用调试器查看代码，这通常是最好的学习方法。

大部分示例都使用了一个叫做 Three.js 的工具集，它是一个杰出的基于 WebGL 的开源引擎。之前，我一直在考虑是给读者详细介绍很多低级别的 WebGL API——这是一个极其灵活、超级强大但是毫无疑问对用户相当不友好的图形绘制系统，还是教给读者如何利用最少的基础知识就可以快速开始构建应用。显然，我做出的最终决定非常明了，那就是让你迅速入门！如果你非常想要知道 WebGL 的底层工作原理和更多底层知识，有许多学习资源可以供你参考。我们在附录 A 中列出了一部分。

WebGL 的开发者已经创立了一个独一无二和不可思议的东西——运行于浏览器中的

3D 内容，在页面中与其他内容一同无缝渲染，整合全球任何地方的数据内容，于是就呈现在你面前一个调色板，用这个调色板你可以构建任何你能够想象到的东西。这并不是零零散散的各个部分的简单组合，而是一个全新的媒介，将会创造一个新的业界局面。WebGL 编程也许一开始并不容易，但我可以承诺，如果你付出了努力，那么一个精彩的全新世界就在等着你。

本书的排版约定

本书使用如下排版约定。

- 楷体：用以表示新术语。
- 等宽代码字体：用于程序清单，以及段落中引用的程序元素，如变量或函数名、数据库、数据类型、环境变量、语句和关键字。
- 加粗的等宽代码字体：展示命令或其他应该由用户输入的文本，也用于代码清单中强调的部分。
- 斜的等宽代码字体：展示应该用用户所提供的值或根据上下文确定的值来替换的文本。



这个图标表示一个提示、建议或者一般的注记。

本书的示例文件

你可以从 GitHub 上下载本书的所有示例代码，网址是：

<http://github.com/tparisi/WebGLBook>

在示例文件中，你可以找到本书中构建的所有应用的完整版，其中包含运行这些应用必需的所有代码。在运行其中的某几个示例之前，你需要下载一些额外的内容，如 3D 模型等，详情请参考根目录中的 README 文件。

代码示例的使用

本书的目标是帮助你完成工作。一般来说，你可以在自己的程序和文档中使用本书中的代码，如果你要复制的不是核心代码，则无须取得我们的许可。例如，你可以在程序中使用本书中的多个代码块，无须获取我们许可。但是，要销售或分发来源于 O'Reilly 图书中的示例的光盘则需要取得我们的许可。通过引用本书中的示例代码来回答问题时，不需要事先获得我们的许可。但是，如果你的产品文档中融合了本书中的大量示例代码，则需要取得我们的许可。

在引用本书中的代码示例时，如果能列出本书的属性信息是最好不过了。属性信息通常包括书名、作者、出版社和 ISBN。例如：“*WebGL: Up and Running* by Tony Parisi (O’Reilly). Copyright 2012 Tony Parisi, 978-1-449-32357-8.”

在使用书中的代码时，如果不确定是否属于合理使用，或是否超出了我们的许可，请通过 permissions@oreilly.com 与我们联系。

我们的联系方式

如果你想就本书发表评论或有任何疑问，敬请联系出版社。

美国：

O’Reilly Media Inc.

1005 Gravenstein Highway North

Sebastopol, CA 95472

中国：

北京市西城区西直门南大街 2 号成铭大厦 C 座 807 室（100035）

奥莱利技术咨询（北京）有限公司

我们还为本书建立了一个网页，其中包含了勘误表、示例和其他额外的信息。你可以通过如下地址访问该网页：

http://oreil.ly/WebGL_UR

关于本书的技术性问题或建议，请发邮件到：

bookquestions@oreilly.com

欢迎登录我们的网站（<http://www.oreilly.com>），查看更多我们的书籍、课程、会议和最新动态等信息。

我们的其他联系方式如下。

Facebook: <http://facebook.com/oreilly>

Twitter: <http://twitter.com/oreillymedia>

YouTube: <http://www.youtube.com/oreillymedia>

致谢

就像 WebGL 本身一样，这本书是很多人努力的成果，其中并不乏许多知名人士的帮助和支持。首先，我想要感谢 O’Reilly 团队。首先是我的编辑 Mary Treseler。新技术书籍一直都很有风险，但 Mary 全速前进最终使书得以出版。她对主题的丰富理解和不断的

鼓励，对于第一次写书的作者来说是非常令人感激的。另外，包括编辑部、发行部和市场营销部门的工作人员，太多的人不能一一列举，他们都非常优秀，让我非常享受写书的过程。

我非常感谢 Giles Thomas (LearningWebGL.com 创始人)、Mike Korczynski、Ray Camden 和 Raffaele Cecco 做出的一流的技术评议。他们详实的评论让我在专业术语上保证了正确性，并且帮我精简了示例代码。最重要的是，在我写作遇到困难时，他们对前几章的积极反馈给了我强大的精神动力。

作为一本编程书籍，其中也包含了大量精美的 3D 图形内容。我想要感谢许多美术设计师授权允许我在本书的示例中使用他们的作品。你可以在每个示例的 HTML 和 JavaScript 文件中以及 README 文件中找到详细的版权信息。我想要特别感谢 Christell Gause, TurboSquid 的技术支持主管，正是由于他的努力，我才可以从多位 TurboSquid 的美术设计师那里获得授权。另外，本书中我编写的示例也离不开来自痴迷于数据的 Theo Armour、资深 3D 开发者 Don Olmstead 和来自 Sunglass.io 的 3D 设计师 Arefin Mohiuddin 的帮助。

WebGL 有幸拥有一个强大的开发社区。我想要感谢 Three.js 的开发者们，包括作者 Ricardo Cabello (网名为 Mr.doob)、代码贡献者 Branislav Ulicny (网名为 AlteredQualia) 和 Tim Knip，感谢他们对于我时常提出的一些幼稚问题的耐心回答以及他们对 Three.js 的热心。我亏欠 Ken Russell 一个人情，他是 WebGL 工作组的主席并在 Google 领导 WebGL 的开发。Ken 不仅构建了 WebGL 这个伟大的产品，还欣然同意为本书作序。

最后，我想要感谢我的朋友和家人。感谢我的老伙伴 Mark Pesce，一起提出 VRML 的“罪人”，他是一位老兵级作家。他对于写作的忘我献身和对新技术的激情，成为在这些年中不断启迪我的源泉。非常感谢我的朋友、有时也是生意伙伴的 Scott Foe，尽管这本书分散了我在我的创业公司的大部分精力，他还是一直支持我写作。最后但同样重要的是，感谢我的家人长期以来给我的精神支持、耐心和充满爱的环境。另外，他们还成为了我的后勤军：我的 10 岁儿子 Lucian，试玩了书中的大部分示例；我的妻子 Marina，让我坚持唯美路线，使得这些拼凑起来的示例看起来外观统一并连贯始终。

目录

第 1 章	WebGL 简介	1
1.1	WebGL——一个技术定义	2
1.2	3D 图形学——入门	4
1.2.1	3D 坐标系	4
1.2.2	网格、多边形和顶点	5
1.2.3	材质、纹理和光源	5
1.2.4	变换与矩阵	6
1.2.5	相机、透视、视口和投影	6
1.2.6	着色器	7
1.3	WebGL 原生 API	8
1.3.1	WebGL 应用结构剖析	9
1.3.2	画布元素与绘制上下文	9
1.3.3	视口	10
1.3.4	Buffer、ArrayBuffer 和类型化数组	10
1.3.5	矩阵	11
1.3.6	着色器	12
1.3.7	绘制图元	13
1.4	本章小结	14
第 2 章	你的第一个 WebGL 程序	15
2.1	Three.js——一个 JavaScript 3D 引擎	15
2.2	建立 Three.js 运行环境	17
2.3	一个简单的 Three.js 网页	17
2.4	一个真实的 3D 示例	20
2.4.1	为场景着色	23
2.4.2	添加纹理映射	24
2.4.3	旋转物体	25
2.4.4	循环重绘和 requestAnimationFrame()	25
2.4.5	让页面贴近生活	26
2.5	本章小结	27
第 3 章	图形	28
3.1	Sim.js——一个轻量级的 WebGL 模拟框架	29
3.2	创建网格	30

3.3	使用材质、纹理和光源	34
3.3.1	光源的种类	35
3.3.2	使用多重纹理创建更具真实感的场景	37
3.3.3	纹理与透明	42
3.4	构建变换层级	42
3.5	创建自定义几何体	46
3.6	点和线的渲染	49
3.6.1	使用粒子系统绘制点	50
3.6.2	线的绘制	52
3.7	编写着色器	53
3.7.1	WebGL 着色器基础	53
3.7.2	Three.js 中的着色器	55
3.8	本章小结	60
第 4 章	动画	61
4.1	动画基础	61
4.1.1	帧动画	61
4.1.2	时间动画	62
4.1.3	插值与补间动画	62
4.1.4	关键帧	63
4.1.5	关节动画	64
4.1.6	蒙皮动画	64
4.1.7	目标变形动画	64
4.2	使用 Tween.js 库来创建补间动画	65
4.2.1	创建一个基本的补间动画	66
4.2.2	带缓动效果的补间动画	68
4.3	为带关节的模型制作关键帧动画	71
4.3.1	载入模型	71
4.3.2	为模型制作动画	73
4.4	材质和光源动画	76
4.5	纹理动画	78
4.6	蒙皮动画和变形动画	80
4.7	本章小结	80
第 5 章	交互	81
5.1	点击检测、拾取和投影	81
	Three.js 中的点击检测	82
5.2	处理鼠标移入和点击	85
5.3	处理拖曳	88

在拖曳中使用补间动画	91
5.4 使用点击点和法线信息	91
5.5 基于相机的交互	92
5.5.1 利用镜头控制制作一个模型浏览器	93
5.5.2 场景漫游	95
5.6 本章小结	96
第 6 章 2D 与 3D 的整合	98
6.1 整合动态 HTML 和 WebGL	99
6.1.1 创建 DIV 元素弹出层	99
6.1.2 利用 2D 屏幕坐标为 3D 物体添加注释	103
6.1.3 为 3D 场景添加背景图片	104
6.2 在 2D 页面上插入 3D 浮层	105
6.3 利用 2D Canvas 创建动态纹理	107
6.4 使用视频作为纹理	115
6.5 渲染动态 3D 文字	119
6.6 WebGL 中的终极整合	121
6.7 本章小结	123
第 7 章 实战 WebGL	124
7.1 如何选择运行库和框架	124
7.2 载入 3D 内容	126
7.2.1 COLLADA: 数字资产交换格式	126
7.2.2 Three.js 中的 JSON 模型文件格式	130
7.2.3 Three.js 二进制模型文件格式	134
7.2.4 压缩 3D 模型	135
7.2.5 Three.js 中的 JSON 场景文件格式	136
7.3 创建 3D 内容	137
7.3.1 从 Blender 中导出 3D 内容	137
7.3.2 把 OBJ 文件转换为 Three.js JSON 文件	139
7.3.3 把 OBJ 文件转换为 Three.js 二进制文件	139
7.3.4 其他软件或格式的转换	139
7.4 浏览器支持度	140
7.4.1 检测浏览器的 WebGL 支持	141
7.4.2 在 Safari 中开启 WebGL 支持	142
7.5 处理丢失上下文事件	143
7.6 WebGL 的安全性	146
7.7 本章小结	149

第 8 章 你的第一个 WebGL 游戏	150
8.1 构建游戏的各个部分	151
8.1.1 相机、角色和控制	152
8.1.2 美术设计	159
8.1.3 模型预览器	161
8.1.4 创建粒子系统	163
8.1.5 添加声音	166
8.2 万物归一	167
8.3 本章小结	180
后记	181
附录 A WebGL 在线资源	183