



华章IT



全面讲解Three.js开发的实用指南，涵盖Three.js的各种功能
通过大量交互式示例，深入探索使用开源的Three.js库创建绚丽三维图形的实用方法和技巧

Web开发技术丛书

Learning Three.js

The JavaScript 3D Library for WebGL, Second Edition

[美] 乔斯·德克森 (Jos Dirksen) 著

杨芬 赵汝达 译

Three.js

开发指南

WebGL的JavaScript 3D库 (原书第2版)



机械工业出版社
China Machine Press

Three.js开发指南

WebGL的JavaScript 3D库

(原书第2版)

[美] 乔斯·德克森 (Jos Dirksen) 著

杨芬 赵汝达 译

Learning Three.js

The JavaScript 3D Library for WebGL, Second Edition



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

Three.js 开发指南: WebGL 的 JavaScript 3D 库 (原书第 2 版) / (美) 乔斯·德克森 (Jos Dirksen) 著; 杨芬, 赵汝达译. —北京: 机械工业出版社, 2017.6
(Web 开发技术丛书)

书名原文: Learning Three.js: The JavaScript 3D Library for WebGL, Second Edition

ISBN 978-7-111-57013-4

I. T… II. ①乔… ②杨… ③赵… III. JAVA 语言—程序设计—指南 IV. TP312.8-62

中国版本图书馆 CIP 数据核字 (2017) 第 104436 号

本书版权登记号: 图字: 01-2017-0491

Jos Dirksen: Learning Three.js: The JavaScript 3D Library for WebGL, Second Edition (ISBN 978-1-78439-221-5).

Copyright © 2015 Packt Publishing. First published in the English language under the title “Learning Three.js: The JavaScript 3D Library for WebGL”.

All rights reserved.

Chinese simplified language edition published by China Machine Press.

Copyright © 2017 by China Machine Press.

本书中文简体字版由 Packt Publishing 授权机械工业出版社独家出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

Three.js 开发指南: WebGL 的 JavaScript 3D 库 (原书第 2 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: 刘立卿

责任校对: 殷虹

印 刷: 三河市宏图印务有限公司

版 次: 2017 年 6 月第 1 版第 1 次印刷

开 本: 186mm × 240mm 1/16

印 张: 18

书 号: ISBN 978-7-111-57013-4

定 价: 69.00 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) 88379426 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzit@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

在最近的几年中，浏览器的功能变得愈发强大，并且成为展现复杂的应用和图形的平台。然而其中的大部分都是标准的二维图形。大多数现代浏览器已经支持 WebGL，不仅可以在浏览器端创建二维应用和图形，而且可以通过 GPU 的功能创建好看并且运行良好的三维应用。

然而，直接使用 WebGL 编程还是很复杂的。编程者需要知道 WebGL 的底层细节，并且学习复杂的着色语言来获得 WebGL 的大部分功能。Three.js 提供一个很简单的关于 WebGL 特性的 JavaScript API，所以用户不需要详细地学习 WebGL，就能创作出好看的三维图形。

Three.js 为直接在浏览器中创建三维场景提供了大量的特性和 API。阅读这本书，读者将通过许多交互案例和示例代码学习 Three.js 提供的各种 API。

本书内容

第 1 章介绍使用 Three.js 的基本步骤。阅读完本章就能立刻创建你的第一个 Three.js 场景，也能直接在浏览器中创建你的第一个三维场景并让它动起来。

第 2 章解释了使用 Three.js 时需要了解的基本组件。将介绍灯光、网格、几何形状、材质和摄像机。阅读本章，读者也将对 Three.js 提供的不同灯光和在场景中使用的摄像机有一个大概的印象。

第 3 章深入介绍可以在场景中使用的不同灯光。本章通过示例讲解了如何使用聚光灯、平行光、环境光、点光源、半球光和区域光。此外，本章还展示了如何使用镜头光晕来影响光源。

第 4 章讲述 Three.js 中可在网格中使用的材质。本章展示了可以设置的所有属性，并且提供了不同的交互示例来阐述 Three.js 中可用的材质。

第 5 章讲述 Three.js 中的几何体。本章将介绍如何在 Three.js 中创建和配置几何体，并且提供了几何体的交互示例，包括平面、圆形、任意图形、长方体、球体、圆柱体、圆环、环状扭结和多面体。

第 6 章接着第 5 章讲解如何使用几何体。本章介绍如何配置和使用 Three.js 提供的更高级的几何体，如凸面体和 Lathe。在本章，你还能学到如何从二维形状拉伸出三维几何

体，并且学会通过二元操作来组合几何体，从而创建出一个新的几何体。

第 7 章介绍从头开始或者通过现有的几何体来创建一个粒子系统。这一章还介绍了如何使用精灵或点云材质来改变单个点的外观。

第 8 章介绍如何通过外部资源导入网格和几何体，以及如何使用 Three.js 内置的 JSON 格式来保存几何体和场景。本章还会解释如何加载不同格式的模型，如 OBJ、DAE、STL、CTM、PLY 等。

第 9 章展示可以让场景动起来的各种动画，包括如何组合使用 Tween.js 库和 Three.js，以及如何使用基于形态和框架的动画模型。

第 10 章延续第 4 章的内容——使用 Three.js 材质。在第 4 章已经初步介绍了材质，本章着重介绍现有的各种材质以及如何把材质应用到网格中。另外，本章还介绍了如何直接使用 HTML5 中的视频元素和画布元素作为材质的输入。

第 11 章展示如何使用 Three.js 对已经渲染过的场景进行后期处理。通过后期处理，你可以将模糊、倾斜移位和着色等效果添加到已经渲染过的场景中。除此之外，本章还介绍了如何创建自己的后期处理效果、自定义的顶点和片段着色器。

第 12 章解释如何将物理效果添加到 Three.js 场景中。通过物理效果，你能检测物体之间的碰撞，使物体对重力有响应，以及应用摩擦力。本章展示了如何用 Physijs JavaScript 库实现这些效果。此外，本章还展示了如何在 Three.js 场景中添加声音。

本书的使用要求

使用本书只需一个能够“玩转”你的例子的文本编辑器（如 Sublime）和能够显示这些例子的现代浏览器。有些例子会需要一个本地的网络服务器，你可以在本书第 1 章中学会如何搭建一个轻量级的网络服务器以使用书中的例子。

面向的读者

这本书对于每一个知道 JavaScript 并且想要在浏览器中创建运行的三维图形的人来说都是很棒的。你不需要知道任何高级的数学知识或 WebGL，所需的只是对 JavaScript 和 HTML 有大致地了解。书中所需要的材料和示例都可以免费下载，所用的工具都是开源的。所以，如果你想创建可以在任何现代浏览器中运行的交互的三维图形，这本书就适合你。

读者反馈

我们一直欢迎读者的反馈。读者反馈让我们知道你对这本书的想法——你是否喜欢。读者反馈对我们非常重要，因为它能帮助我们开发读者真正想要的主题。

你可以简单地发一封邮件到 feedback@packtpub.com，在邮件标题中标明本书的标题。

如果你在某个主题方面十分专业并且有兴趣写书，可以在 www.packtpub.com/authors 查看我们的作者指南。

客户支持

现在你已经拥有一本 Packet 书了，我们有很多东西可以帮助你，希望你能从这次购买中得到最大的价值。

下载源码

你可以访问 <http://www.packtpub.com> 并通过你的账号下载所购买的 Packt 图书的所有示例源码。如果你是在别的地方购买的图书，也可以访问 <http://www.packtpub.com/support> 并注册，我们会直接将示例代码通过电子邮件发送给你。

下载本书的彩图

我们同样提供了一个 PDF 文件，其中保存了书中用到的所有截图或图表的彩色图片。这些彩图会帮助你更好地理解输出结果的变化。你可以访问 https://www.packtpub.com/sites/default/files/downloads/2215OS_Graphics.pdf 下载这个文件。

勘误表

尽管我们已经尽量保证内容的准确性，但错误总是难免的。如果你在本书中找到了错误，无论是文字错误或代码错误，请告诉我们，我们将非常感激。你的帮助，不仅可以让其读者避免遇到同样的错误，还有助于我们在本书的后续版本中提供更高质量的信息。如果你发现了任何一处错误，请通过访问 <http://www.packtpub.com/submit-errata> 来提交：选择书名，点击 Errata Submission Form 链接，输入你提交的勘误的详情。一旦你的勘误验证通过，你的提交将被接受，勘误会上传到我们的网站或添加到任何与这个勘误相关的勘误表中。

如果你想查阅之前提交的勘误表，请访问 <https://www.packtpub.com/books/content/support>，在搜索框中输入书名，相关信息会显示在“勘误表”部分。

关于盗版

互联网上版权材料的盗版是所有媒体一直面临的问题。在 Packet，我们对版本和许可证的保护十分重视。如果你在互联网上看到了任何形式的任何一份我们作品的非法版本，请立即向我们提供地址或网站名称，以便我们采取补救措施。

请通过 copyright@packtpub.com 联系我们，附上嫌疑盗版作品的链接。

你能帮助我们的作者维护知识产权，我们深表感谢。

问题

如果对本书有任何问题，可以通过 question@packtpub.com 与我们联系，我们会尽最大的努力来解决。

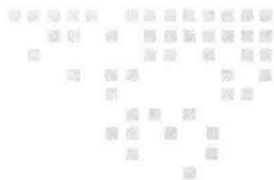
目 录 Contents

前言	
第 1 章 使用 Three.js 创建你的第一个三维场景 1	
1.1 准备工作..... 4	
1.2 获取源码..... 5	
1.2.1 通过 Git 获取代码仓库..... 5	
1.2.2 下载并解压缩档案文件..... 5	
1.2.3 测试示例..... 6	
1.3 搭建 HTML 框架..... 9	
1.4 渲染并查看三维对象..... 10	
1.5 添加材质、光源和阴影效果..... 14	
1.6 让你的场景动起来..... 16	
1.6.1 引入 requestAnimationFrame() 方法..... 16	
1.6.2 旋转立方体..... 17	
1.6.3 弹跳球..... 18	
1.7 使用 dat.GUI 简化试验流程..... 19	
1.8 场景对浏览器的自适应..... 21	
1.9 总结..... 22	
第 2 章 构建 Three.js 场景的基本组件 ... 23	
2.1 创建场景..... 23	
2.1.1 场景的基本功能..... 24	
2.1.2 给场景添加雾化效果..... 29	
2.1.3 使用 overrideMaterial 属性..... 30	
2.2 几何体和网格..... 31	
2.2.1 几何体的属性和方法..... 31	
2.2.2 网格对象的属性和方法..... 36	
2.3 选择合适的摄像机..... 40	
2.3.1 正交投影摄像机和透视投影摄像机..... 40	
2.3.2 将摄像机聚焦在指定点上..... 44	
2.4 总结..... 45	
第 3 章 学习使用 Three.js 中的光源 ... 46	
3.1 Three.js 中不同类型的光源..... 46	
3.2 基础光源..... 47	
3.2.1 THREE.AmbientLight..... 47	
3.2.2 THREE.PointLight..... 51	
3.2.3 THREE.SpotLight..... 54	
3.2.4 THREE.DirectionalLight..... 58	
3.3 特殊光源..... 60	
3.3.1 THREE.HemisphereLight..... 60	
3.3.2 THREE.AreaLight..... 61	
3.3.3 镜头光晕..... 64	

3.4 总结	66	6.3.1 THREE.ExtrudeGeometry	117
第4章 使用 Three.js 的材质	67	6.3.2 THREE.TubeGeometry	119
4.1 理解材质的共有属性	68	6.3.3 从 SVG 拉伸	120
4.1.1 基础属性	68	6.3.4 THREE.ParametricGeometry	122
4.1.2 融合属性	69	6.4 创建三维文本	124
4.1.3 高级属性	70	6.4.1 渲染文本	124
4.2 从简单的网格材质开始	70	6.4.2 添加自定义字体	126
4.2.1 THREE.MeshBasicMaterial	71	6.5 使用二元操作组合网格	127
4.2.2 THREE.MeshDepthMaterial	73	6.5.1 subtract 函数	129
4.2.3 联合材质	75	6.5.2 intersect 函数	132
4.2.4 THREE.MeshNormalMaterial	76	6.5.3 union 函数	133
4.2.5 THREE.MeshFaceMaterial	78	6.6 总结	134
4.3 高级材质	81	第7章 粒子、精灵和点云	135
4.3.1 THREE.MeshLambertMaterial	81	7.1 理解粒子	135
4.3.2 THREE.MeshPhongMaterial	82	7.2 粒子、THREE.PointCloud 和 THREE.PointCloudMaterial	138
4.3.3 用 THREE.ShaderMaterial 创建 自己的着色器	83	7.3 使用 HTML5 画布样式化粒子	140
4.4 线性几何体的材质	89	7.3.1 在 THREE.CanvasRenderer 中使用 HTML5 画布	140
4.4.1 THREE.LineBasicMaterial	89	7.3.2 在 WebGLRenderer 中使用 HTML5 画布	142
4.4.2 THREE.LineDashedMaterial	91	7.4 使用纹理样式化粒子	144
4.5 总结	92	7.5 使用精灵贴图	149
第5章 学习使用几何体	93	7.6 从高级几何体创建 THREE.Point Cloud	153
5.1 THREE.js 提供的基础几何体	94	7.7 总结	155
5.1.1 二维几何体	94	第8章 创建、加载高级网格和 几何体	156
5.1.2 三维几何体	102	8.1 几何体组合与合并	156
5.2 总结	112	8.1.1 对象组合	156
第6章 高级几何体和二元操作	113	8.1.2 将多个网格合并成一个网格	158
6.1 THREE.ConvexGeometry	113		
6.2 THREE.LatheGeometry	115		
6.3 通过拉伸创建几何体	116		

8.1.3	从外部资源中加载几何体	160			
8.1.4	以 Three.js 的 JSON 格式保存 和加载	161			
8.1.5	使用 Blender	165			
8.1.6	导入三维格式文件	169			
8.2	总结	179			
第 9 章 创建动画和移动摄像机 180					
9.1	基础动画	180			
9.1.1	简单动画	181			
9.1.2	选择对象	182			
9.1.3	使用 Tween.js 实现动画	184			
9.2	使用摄像机	186			
9.2.1	轨迹球控制器	187			
9.2.2	飞行控制器	189			
9.2.3	翻滚控制器	190			
9.2.4	第一视角控制器	191			
9.2.5	轨道控制器	192			
9.3	变形动画和骨骼动画	193			
9.3.1	用变形目标创建动画	195			
9.3.2	用骨骼和蒙皮创建动画	198			
9.4	使用外部模型创建动画	200			
9.4.1	使用 Blender 创建骨骼动画	201			
9.4.2	从 Collada 模型加载动画	203			
9.4.3	从雷神之锤模型中加载动画	204			
9.5	总结	205			
第 10 章 加载和使用纹理 206					
10.1	将纹理应用于材质	206			
10.1.1	加载纹理并应用到网格	206			
10.1.2	使用凹凸贴图创建褶皱	210			
10.1.3	使用法向贴图创建更加细 致的凹凸和褶皱	211			
10.1.4	使用光照贴图创建阴影效果	212			
10.1.5	使用环境贴图创建反光效果	214			
10.1.6	高光贴图	218			
10.2	纹理的高级用途	220			
10.2.1	自定义 UV 映射	220			
10.2.2	重复纹理	222			
10.2.3	在画布上绘制图案并作为 纹理	224			
10.2.4	将视频输出作为纹理	227			
10.3	总结	229			
第 11 章 自定义着色器和后期处理 230					
11.1	配置 Three.js 以进行后期处理	230			
11.2	后期处理通道	233			
11.2.1	简单后期处理通道	233			
11.2.2	使用掩码的高级效果组合器	239			
11.2.3	使用 THREE.ShaderPass 自 定义效果	242			
11.3	创建自定义后期处理着色器	248			
11.3.1	自定义灰度图着色器	248			
11.3.2	自定义位着色器	251			
11.4	总结	253			
第 12 章 在场景中添加物理效果和 声音 254					
12.1	创建基本的 Three.js 场景	254			
12.2	材质属性	259			
12.3	基础图形	260			
12.4	使用约束限制对象的移动	265			
12.4.1	使用 PointConstraint 限制 对象在两点间移动	266			

12.4.2 使用 HingeConstraint 创建类似门的约束.....	267	创建类似球销的约束.....	270
12.4.3 使用 SliderConstraint 将移动限制在一个轴上.....	269	12.4.5 使用 DOFConstraint 实现细节的控制.....	272
12.4.4 使用 ConeTwistConstraint		12.4.6 在场景中添加声源.....	276
		12.5 总结	277



使用 Three.js 创建你的第一个三维场景

现代浏览器直接通过 JavaScript 就可以实现非常强大的功能。使用 HTML5 的标签可以很容易地添加语音和视频，而且在 HTML5 提供的画布上可以添加具有交互功能的组件。现在，现代浏览器也开始支持 WebGL。通过 WebGL 可以直接使用显卡资源来创建高性能的二维和三维图形，但是直接使用 WebGL 编程来创建三维动画场景十分复杂而且还容易出问题。使用 Three.js 库可以简化这个过程。Three.js 带来的好处有以下几点：

- 创建简单和复杂的三维几何图形。
- 在三维场景下创建动画和移动物体。
- 为物体添加纹理和材质。
- 使用各种光源来装饰场景。
- 通过三维模型软件加载物体。
- 为三维场景添加高级的后期处理效果。
- 使用自定义的着色器。
- 创建点云（即粒子系统）。

通过几行简单的 JavaScript 代码，你可以创建从简单三维模型到类似图 1.1（在浏览器中访问 <http://www.vill.ee/eye>）所示的具有真实感的场景。

本章将会通过示例来阐述 Three.js 的工作原理，但是不会对其中的技术细节进行深入探究，这些细节我们将会在后面的章节中介绍。本章主要涵盖以下几方面：

- 使用 Three.js 所需的工具
- 下载本书所需的源码和示例

- ❑ 创建第一个 Three.js 场景
- ❑ 使用材质、光源和动画来完善第一个场景
- ❑ 引入辅助库以统计和控制场景

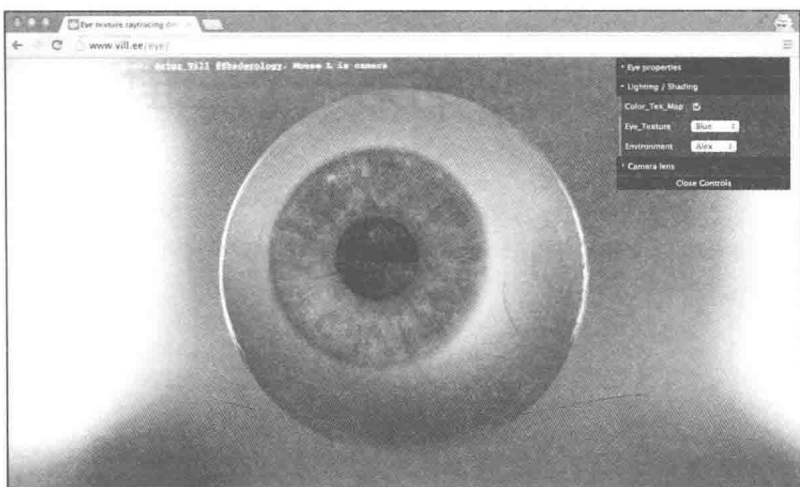


图 1.1

首先将会对 Three.js 进行简单的介绍, 然后再讲解第一个示例及其代码。在开始之前我们先来看下当前主流浏览器对 WebGL 的支持情况, 具体见表 1.1。

表 1.1

浏览器	支持情况
Mozilla Firefox	4.0 版本以后开始支持
Google Chrome	9.0 版本以后开始支持
Safari	5.1 版本以及那些安装在 Mac OS X 的山狮版、狮子版和雪豹版操作系统上的版本支持。确保在 Safari 中打开了 WebGL 选项, 在 Preferences Advanced 菜单选项中选中 Show develop menu in menu bar。然后选中菜单项 Develop Enable WebGL
Opera	12.00 版本以后开始支持。为了让 Opera 支持 WebGL, 你还需要打开 opera:config 文件设置 WebGL 的值, 将 Enable Hardware Acceleration 设置为 1, 然后重启浏览器
Internet Explorer	Internet Explorer 是唯一一个很长时间都不支持 WebGL 的浏览器, 从 IE11 才开始支持 WebGL

除了 IE 的一些低版本浏览器, 基本所有的现代浏览器都支持 Three.js。如果想在低版本的 IE 浏览器上运行 Three.js, 你还需要做额外的操作。对于 IE10 和更低的版本, 你可以安装 iewebgl 插件, 下载地址为 <https://github.com/iewebgl/iewebgl>。

Three.js 还可以在移动设备上运行, 不同设备对于 Three.js 的支持情况如表 1.2 所示。

表 1.2

设 备	支持情况
Andriod	Andriod 原生的浏览器设备是不支持 WebGL 的，而且对于 HTML5 的高级特性也不支持。如果想在 Andriod 上运行 WebGL，你还需要安装最新的移动版本的 Chrome、Firefox 或者 Opera 浏览器
IOS	从 IOS 8 开始，IOS 设备就开始支持 WebGL。IOS 的 Safari 版本 8 是对 WebGL 支持最好的
Windows mobile	8.1 版本以后开始支持

使用 WebGL 能够创建出具有交互性的 3D 模型，而且这些模型在电脑和手机设备上都能够很好地运行。



本书主要介绍 Three.js 提供的基于 WebGL 的渲染器。但是 Three.js 也提供了基于 CSS-3D 的渲染器，使用其 API 能够很容易地创建出三维场景，而且这个渲染器的优点在于几乎所有手机和电脑上的浏览器都支持 CSS-3D，并且可以在三维空间内渲染 HTML 元素。第 7 章会进一步介绍如何使用 CSS-3D。

通过本章的学习，你可以直接创建第一个三维场景，而且这个场景可以在上述的所有浏览器中运行。目前我们不会介绍太多 Three.js 的复杂特性，但是在本章结束的时候你能够创建出如图 1.2 所示的场景。

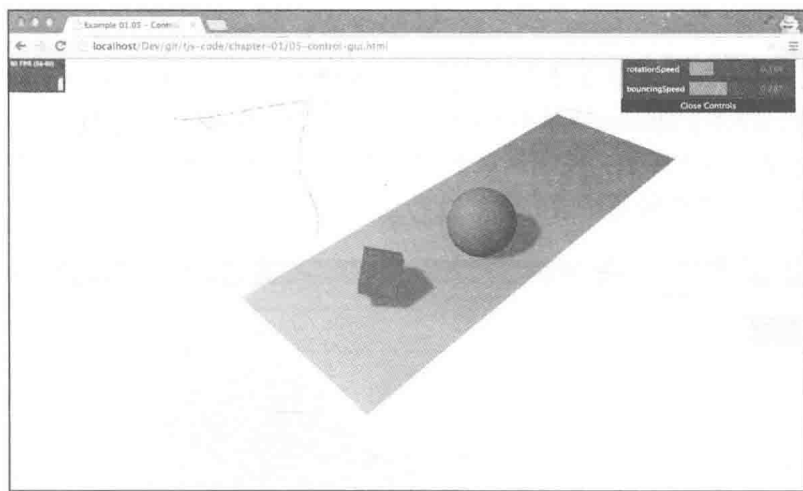


图 1.2

通过创建这个场景，你可以学到 Three.js 的基础知识，并可以创建第一个动画。在开始实现这个示例之前，我们会先介绍几个工具，这些工具可以帮助你更方便地使用 Three.js。我们还会介绍如何下载本书所用到的示例。

1.1 准备工作

Three.js 是一个 JavaScript 库，所以你只需要一个文本编辑器和支持 Three.js 的浏览器就可以使用 Three.js 来创建 WebGL 应用。这里推荐如下几款 JavaScript 文本编辑器：

- ❑ **WebStorm**: WebStorm 是 JetBrains 公司旗下的一款 JavaScript 编辑工具。它支持代码补全、自动部署和代码检测功能。除此之外，WebStorm 还支持 GitHub 和其他各种版本控制器。读者可从 <http://www.jetbrains.com/webstorm> 下载一个试用版本。
- ❑ **Notepad++**: Notepad++ 是 Windows 操作系统下的通用文本编辑器，它支持各种编程语言语法高亮度显示，而且可以很容易地对 JavaScript 进行布局和格式化。Notepad++ 的下载地址为 <http://notepad-plus-plus.org/>。
- ❑ **Sublime Text**: Sublime Text 是一款对 JavaScript 支持非常好的文本编辑器。除此之外，Sublime Text 的一大亮点是支持多重选择——同时选择多个区域，然后同时进行编辑。这些功能提供了一个很好的 JavaScript 编程环境。Sublime Text 是一个收费闭源软件，下载地址为 <http://www.sublimetext.com/>。

除此之外，还有很多可以编写 JavaScript 进而创建 Three.js 应用的开源和商用编辑器。还有一款基于云的代码编辑平台 Cloud9，网址是 <http://c9.io>，该平台可以连接 GitHub 账号，由此可以直接获取本书相关的代码和示例。



除了使用这些文本编辑器来运行本书相关的代码和示例，Three.js 自身也提供了在线编辑器，访问地址为 <http://threejs.org/editor>。使用该编辑器，可以用图形化方法创建 Three.js 场景。

虽然现代浏览器基本都支持 WebGL 并能运行 Three.js 应用，但是本书所采用的是 Chrome 浏览器。因为 Chrome 是对 WebGL 支持最好的浏览器，并且拥有强大的 JavaScript 代码调试功能。如图 1.3 所示，使用调试器的断点和控制台功能可以快速地定位问题。在本书中，还会进一步介绍各种调试的方法和技巧。

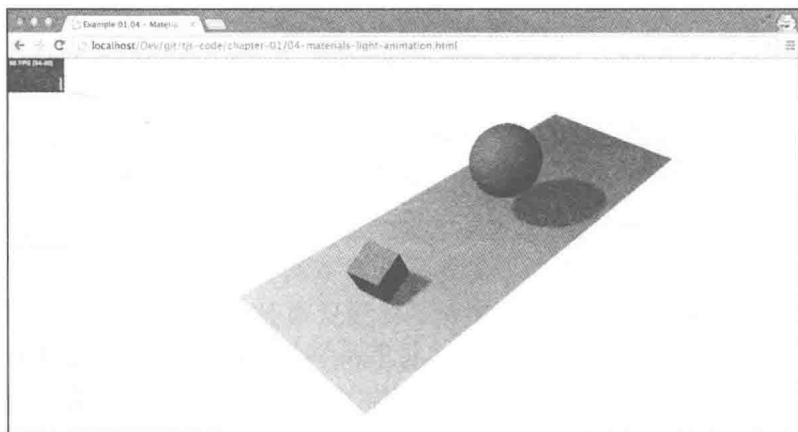


图 1.3

Three.js 就介绍到此，下面就开始获取源码并创建我们的第一个场景。

1.2 获取源码

本书所有的源码都可以从 GitHub (<https://github.com/>) 获取。GitHub 是基于 Git 的在线代码仓库，你可以使用它来存储、访问源码和进行版本控制。获取 GitHub 上源码的方式有以下两种：

- ❑ 通过 Git 获取代码仓库
- ❑ 下载并解压缩档案文件

接下来会详细讲解这两种方式。

1.2.1 通过 Git 获取代码仓库

Git 是开源、分布式的版本控制系统，本书的示例都是基于 Git 进行创建和版本管理的，GitHub 仓库的访问地址为 <https://github.com/josdirksen/learning-threejs>。

如果你已经在操作系统中安装了 Git 的客户端，那么就可以使用 `git` 命令来克隆示例的代码仓库。如果你还没有安装，可以访问 <http://git scm.com> 下载，或者使用 GitHub 公司自己为 Mac 或 Windows 系统提供的客户端。在安装完 Git 客户端后，打开控制台并在你想要存储代码的目录中运行如下的命令：

```
# git clone https://github.com/josdirksen/learning-threejs
```

如图 1.4 所示，代码就开始下载了。

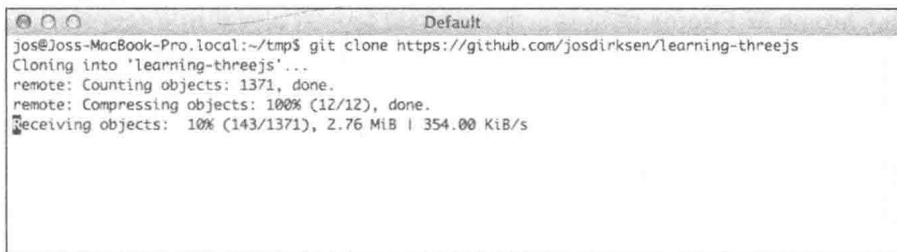


图 1.4

下载完毕后，在 `learning-three.js` 文件夹中会看到本书所用的所有的示例。

1.2.2 下载并解压缩档案文件

如果你不想使用 Git 的方式从 GitHub 上获取源码，那么还可以在 GitHub 上下载档案文件。在浏览器上访问 <https://github.com/josdirksen/learning-threejs>，点击右侧的 Download ZIP 按钮。如图 1.5 所示。

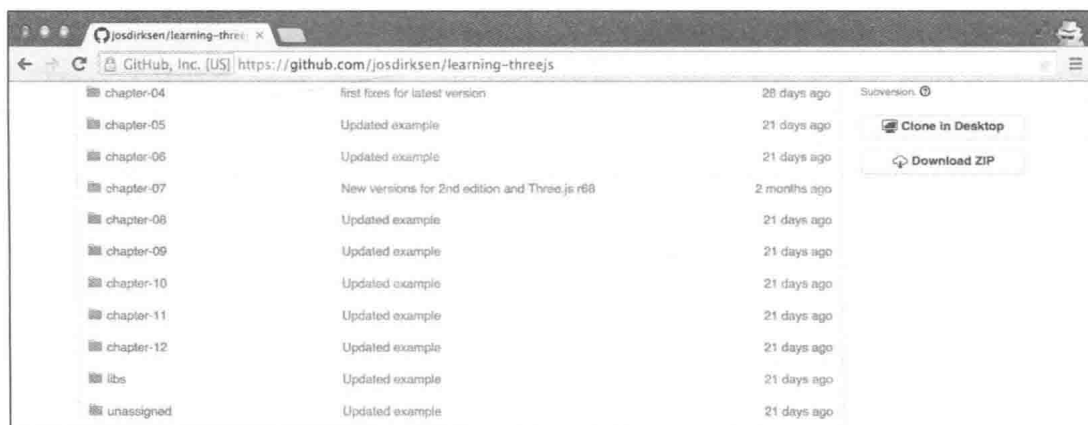


图 1.5

解压文件到指定的目录，这样就可以获取所有的示例了。

1.2.3 测试示例

现在你已经下载或者克隆了示例源码，下面我们就来测试示例是否能够正常工作，进而了解工程的目录结构。本书的示例是按照章节进行组织的。如果要运行本书的示例，你可以使用浏览器打开 HTML 文件，或者安装本地 Web 服务器。第一种方法对于简单的示例是可行的，但是如果示例中需要下载外部资源时，比如模型或者纹理图像，那么仅仅使用浏览器打开 HTML 文件是行不通的。这个时候，我们就需要本地 Web 服务器来确保外部资源正确加载。接下来我们会介绍几种安装本地服务器的方式，如果你无法安装本地服务器但使用 Chrome 或 Firefox 浏览器，那么我们也会介绍如何禁用安全性检测来运行示例。

接下来，我们将会介绍几种安装本地服务器的方式，依据系统的配置，你可以选择最合适的方式。

1. 适用于 Unix/Mac 系统的基于 Python 的 Web 服务器

大部分的 Unix/Linux/Mac 系统默认安装了 Python，在示例源码目录中运行如下的命令就可以将本地 Web 服务器启动起来了。

```
> python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
```

在你下载了源码的目录中做这些。

2. 基于 NPM 的 Web 服务器

如果你已经在使用 Node.js 了，那么你肯定已经安装了 NPM。使用 NPM，有两个方式可以快速地搭建本地 Web 服务器：

第一种方式是使用 http-server 模块，如下所示：


```
> npm install -g http-server
> http-server
Starting up http-server, serving ./ on port: 8080
Hit CTRL-C to stop the server
```

第二种方式是使用 simple-http-server，如下所示：

```
> npm install -g simple-http-server
> nserver
simple-http-server Now Serving: /Users/jos/git/Physijs at http://
localhost:8000/
```

但是第二种方式的缺点在于无法自动地显示目录列表，而第一种方式是可以的。

3. Mac/Windows 上的轻量级服务器——Mongoose

如果你还没有安装 Python 或者 NPM，那么还有一个简单、轻量级的 Web 服务器——Mongoose。首先，从 <https://code.google.com/p/mongoose/downloads/list> 下载你的系统所支持的二进制安装文件。如果你使用的是 Windows 系统，那么将下载好的二进制文件复制到示例所在的目录，双击即可启动一个运行于该目录下的 Web 服务器。

对于其他的操作系统，则须将下载的二进制文件复制到指定的目录中，但是启动的方式不是双击，而是通过命令行的方式。在这两种情况下都会在 8080 端口启动一个本地 Web 服务器。本书的示例目录如图 1.6 所示。

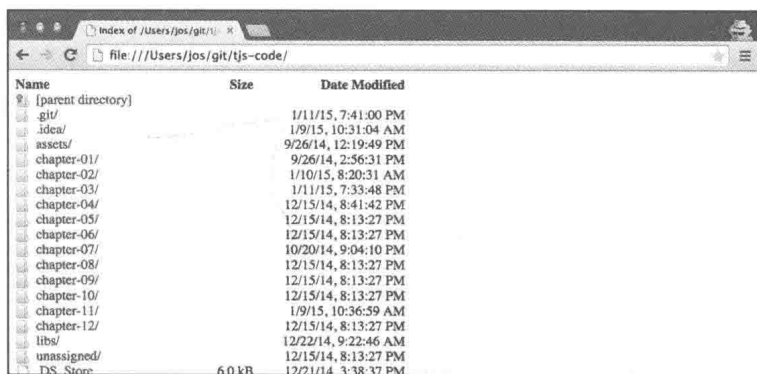


图 1.6

点击各章节的目录就可以获取相应章节的示例。在本书讲解提到某个示例时会指出示例的名称和所在的文件夹，这样你就可以找到该示例并查看源代码。

4. 禁用 Firefox 和 Chrome 的安全性检测

如果你使用的是 Chrome 浏览器，有种方式可以禁用安全性设置，这样就可以在没有 Web 服务器的情况下查看示例。但是需要以如下的方式启动浏览器：