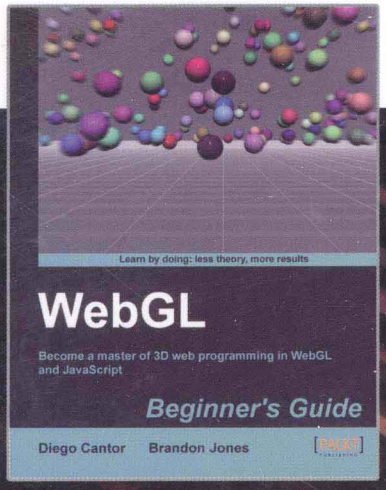
WebGL Beginner's Guide

WebGL编程指南

Diego Cantor Brandon Jones 著
李 强 译



清华大学出版社

WebGL 编程指南

Diego Cantor　Brandon Jones　著

李　强　译

清华大学出版社

内 容 简 介

本书详细阐述了与 WebGL 程序设计相关的基本解决方案，主要包括 WebGL 概述，渲染几何体，光照，相机，实现方案，颜色、深度测试以及 Alpha 混合，纹理，拾取操作，整合方案以及高级话题等内容。此外，本书还提供了相应的示例、代码以及伪代码，以帮助读者进一步理解相关方案的实现过程。

本书可以作为高等院校计算机及相关专业的教材和教学参考书，也可作为相关开发人员的自学教材和参考手册。

图书在版编目（CIP）数据

WebGL 编程指南/（美）坎特（Cantor, D.），（美）琼斯（Jones, B.）著；李强译. —北京：清华大学出版社，2013.12

书名原文：WebGL Beginner's Guide

ISBN 978-7-302-34890-0

I. ①W…　II. ①坎…　②琼…　③李…　III. ①Web 页制作工具-程序设计　IV. ①TP393.092

中国版本图书馆 CIP 数据核字（2013）第 311157 号

责任编辑：钟志芳
封面设计：刘　超
版式设计：文森时代
责任校对：王　云
责任印制：何　芊

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn

印 装 者：北京鑫海金澳胶印有限公司
经　　销：全国新华书店
开　　本：185mm×230mm　　**印　　张**：18.75　　**字　　数**：384 千字
版　　次：2013 年 12 月第 1 版　　**印　　次**：2013 年 12 月第1 次印刷
印　　数：1～4000
定　　价：59.00 元

产品编号：053952-01

译 者 序

WebGL 完美地解决了现有的 Web 交互式三维动画的两个问题：第一，作为一种新兴的 Web 技术，WebGL 向浏览器添加了基于硬件加速的 3D 图形功能，且用户无须安装附加软件；第二，它利用底层的图形硬件加速功能进行图形渲染，并通过统一的、标准的、跨平台的 OpenGL 接口加以实现。

WebGL 是一种 3D 绘图标准，该标准允许把 JavaScript 和 OpenGL ES 2.0 结合在一起，可以为 HTML 5 canvas 提供 GPU 加速功能，Web 开发人员可以借助 PC 的显卡在浏览器中更加流畅地展示 3D 场景和模型，还能创建复杂的导航和数据视觉化系统，它还具有复杂 3D 结构的网站页面，甚至可以用来设计 3D 网页游戏。

WebGL 和 OpenGL 一样来自 Khronos Group，而且免费开放。微软 Silverlight 3.0、Adobe Flash Player 11 也都已经支持 GPU 加速，但它们都是私有的、不透明的。作为一项开放的 Web 标准，Mozilla、Apple、Google、Opera 等公司和组织都是其中的成员，即这一标准的制定者和积极倡导者。这些成员与 Khronos 公司通力合作，创建一种多平台环境可用的 WebGL 标准。

在本书的翻译过程中，除刘鹏外，王晓晓、梁洪娇、李亚楠、姚楷锋、孙年果、丁妍、李中卉、王典、胡子文、朱利平、刘凌、史云龙、李保金、皮雄飞、孙健、王巍、程聪、李海俊等人也参与了部分翻译工作，在此一并表示感谢。

限于译者的水平，译文中难免有错误和不妥之处，恳请广大读者批评指正。

译 者

前　言

作为一种新兴的 Web 技术，WebGL 向浏览器添加了基于硬件加速的 3D 图形功能，且用户无须安装附加软件。WebGL 源自 OpenGL，因而对 Web 程序开发而言，这带来了全新的 3D 图形程序设计理念，即使对于资深的 Web 开发人员来说，这一说法亦不为过。

本书涵盖了大量示例，进而阐述 WebGL 的学习方式。其中，各章节分别对 3D 图形程序设计的某一重要问题加以讨论，并辅以练习以对相关理论予以直接测试。

关于 WebGL 的学习方式，本书提供了一幅清晰的线路图。各章均提供了相应的学习目标，随后是对相关内容的深入讨论。除此之外，本书示例丰富，并引入了基于 WebGL 的最新理念，包括绘制机制、颜色、纹理、转换操作、帧缓冲区、光照、对象表面、几何体对象等。在 WebGL 环境下，各章示例使理论与实际得以有机结合，进而全面提升读者的 3D 图形程序设计水平。本书使用 WebGL 和 JavaScript 语言，并涵盖了与 3D Web 图形应用程序相关的大量信息，因而其实用性和重要性毋庸置疑。

本书内容

第 1 章引入了 HTML 5 canvas 元素，阐述了与其相关的 WebGL 上下文环境，并于随后讨论了 WebGL 应用程序的基本结构。作为 WebGL 功能展示，本书将对车辆仓库应用程序予以分析，该示例体现了 WebGL 应用程序中的各方面内容。

第 2 章介绍了 WebGL API，并以此定义、处理和渲染对象。同时，本章还讲述了基于 AJAX 和 JSON 的异步几何体对象加载方式。

第 3 章讲解 WebGL 环境中的 ESSL 着色器语言，以及 ESSL 着色器中 WebGL 场景的光照实现方案。另外，本章还将深入分析着色器机制和反射光照模型背后的理论知识，并将其与多个示例加以结合。

第 4 章探讨了矩阵代数，并在 WebGL 中构建和操控相机对象，其相关内容涉及 WebGL 场景中的透视和法线矩阵，描述了此类矩阵与 ESSL 着色器之间的传递方式，进而将其应用于各顶点上，并展示了与 WebGL 相机构建方式相关的多个示例。

第 5 章对矩阵应用进行了适当的扩展，进而执行场景元素的转换行为（如移动、旋转以及缩放操作）。同时，本章还引入了矩阵栈这一概念，据此可针对场景中的各对象执行并维护独立转换。另外，本章还通过矩阵栈和 JavaScript 计时器讨论了多种动画技术，

并通过具体示例予以展示。

第 6 章深入分析了 ESSL 中的颜色应用，其中包括 WebGL 场景中的多光源的定义和操作方式、深度测试概念、Alpha 混合机制，以及基于此类特性的透明对象的创建方式，并辅以多个示例对相关概念予以说明。

第 7 章解释了 WebGL 场景中的纹理的创建、管理以及映射方式，纹理坐标以及纹理贴图机制。相关示例展示了多种贴图技术。另外，本章还阐述了多重纹理和立方体贴图的应用方式。

第 8 章描述了拾取操作的简单实现方案，该术语体现了用户与场景对象的选取和交互方式。拾取操作计算鼠标的单击坐标，并确定用户是否在 canvas 渲染对象上执行了单击操作。该方案的框架通过多个回调钩子函数加以定义，进而实现特定逻辑的应用程序。同样，本章依然提供了多个演示示例。

第 9 章回顾了示例代码的框架结构，并对第 1 章提及的虚拟车辆仓库应用程序予以回顾，还进行了适当的扩展。作为学习示例，该应用程序显示了 Blender 模型于 WebGL 场景之间的导入方式，以及如何创建 ESSL 着色器以支持 Blender 中的材质数据。

第 10 章探讨了多个高级话题，如后处理技术、点精灵对象、法线贴图以及光线跟踪技术，并辅以对应示例。通过阅读本书，读者可为日后的学习打下坚实的基础。

本书适用读者

本书适用于对 3D Web 开发感兴趣的 JavaScript 使用者，读者应理解基本的 DOM 对象模型、jQuery 库、AJAX 以及 JSON 等技术，但此类技术并非必需。同时，读者无须具备 WebGL 知识。此外，本书假设读者了解基本的线代数运算。

本书结构

本书包含了大量的示例，以对所学内容进行多方尝试。另外，本书对于各种信息采取了不同的叙述方式。例如，代码通常描述为“通过某一支持的浏览器打开 ch1_Canvas.html 文件”。

对应代码结构如下所示。

```
<!DOCTYPE html>
<html>
<head>
  <title> WebGL Beginner's Guide - Setting up the canvas </title>
  <style type="text/css">
```

```
  canvas {border: 2px dotted blue;}
  </style>
</head>
<body>
<canvas id="canvas-element-id" width="800" height="600">
Your browser does not support HTML5
</canvas>
</body>
</html>
```

需要着重强调的代码则采用黑体表示，如下所示。

```
<!DOCTYPE html>
<html>
<head>
  <title> WebGL Beginner's Guide - Setting up the canvas </title>
  <style type="text/css">
  canvas {border: 2px dotted blue;}
  </style>
</head>
<body>
<canvas id="canvas-element-id" width="800" height="600">
Your browser does not support HTML5
</canvas>
</body>
</html>
```

命令行输入或输出内容则通过下列方式编写。

```
--allow-file-access-from-files
```

图标表示较为重要的概念，而图标则表示提示或相关操作技巧。

资源下载

读者可访问 http://www.packtpub.com 下载本书中的示例代码文件；或者访问 http://www.packtpub.com/support，经注册后可直接通过邮件方式获取相关文件。

除此之外，读者还可获取 PDF 格式文件，文件中包含了本书涉及的截图和图标，这将有助于读者理解不同输出结果之间的差异。读者可访问 http://www.packtpub.com/sites/default/files/downloads/1727_images.pdf 下载该文件。

勘误表

尽管我们在最大程度上做到尽善尽美，但错误依然在所难免。如果读者发现谬误之处，无论是文字错误抑或是代码错误，还望不吝赐教。这对于其他读者以及本书的再版工作，将具有十分重要的意义。对此，读者可访问 http://www.packtpub.com/support，选取对应书籍，单击 errata submission form 超链接，并输入相关问题的详细内容。经确认后，填写内容将被提交至网站，或添加至现有勘误表中（位于该书籍的 Errata 部分）。同时，读者还可访问 http://www.packtpub.com/support 查看当前勘误表。

版权须知

一直以来，互联网上的版权问题从未间断，Packt Publishing 出版社对此类问题异常重视。若读者在互联网上发现本书任意形式的副本，请告知网络地址或网站名称，我们将对此予以处理。

关于盗版问题，读者可发送邮件至 copyright@packtpub.com。

对于作者的爱护，我们表示衷心的感谢，并于日后向读者呈现更为精彩的作品。

问题解答

若读者对本书有任何疑问，均可发送邮件至 questions@packtpub.com，我们将竭诚为您服务。

本书作者及审校人员

Diego Hernando Cantor Rivera 出生于哥伦比亚波哥大，目前是一名软件工程师。Diego 于 2002 年完成了其大学学业，从事计算机视觉系统开发工作，即与计算机交互的视觉跟踪技术，并于 2005 年获取了计算机工程硕士学位，其专业方向为计算机软件体系结构和医学图像处理技术。期间，作为一名实习生，Diego 曾分别参与了法国里昂 CREATIS 图形处理实验室，以及澳大利亚布里斯班 E-Health 研发中心的实习工作。目前，Diego 正在位于加拿大伦敦的西安大略大学攻读博士学位，并负责神经外科增强现实系统的研发工作。

2010 年早期，当 WebGL 技术首次出现于浏览器时，Brandon Jones 即从事该领域的研发工作，Brandon 具有 8 年的 Web 开发经验，对实时图形学抱有极大的热情，并致力

于二者的完美结合。目前，Brandon 在摩托罗拉移动公司从事 HTML 5 研发工作。

Paul Brunt 具有 10 年的 Web 开发经验，并于初期从事电子商务方面的开发工作。强大的程序设计背景以及扎实的数学知识，使得他在 HTML 5 产生初期即有机会从事媒体技术方面的开发工作，即基于 Web 技术的游戏研发。在 HTML 5 刚出现时，Paul 通过 JavaScript 技术开发游戏以及其他应用程序（SVG 扩展应用、canvas 以及高速 JavaScript 引擎开发），其中包括称为 Berts Breakdown 的、基于概念平台的游戏展示模型。

在计算机技术、Blender 以及实时图形学的基础上，WebGL 的出现催生了 GLGE 的发展。2009 年，当 WebGL 尚处于雏形期时，Paul 即从事 GLGE 方面的研发工作，并将目光关注于在线游戏开发。

在离开 GLGE 后，Paul 从事 WebGL 框架等项目的研发工作。2010 年，他将 JigLib 物理库整合至 JavaScript 中，并首次在浏览器中展示了 2D 物理效果。

Dan Ginsburg 是 Upsample Soft ware、LLC 的奠基人，该公司提供了 3D 图形以及 GPU 计算技术方面的专业咨询服务。除此之外，Dan 还是 *OpenGL ES 2.0 Programming Guide* 和 *OpenGL Programming Guide* 的联合作者。Dan 分别在伍斯特理工学院和本特利大学获得计算机科学理学士和工商管理学硕士学位。

Andor Salga 毕业于圣力嘉学院并获得软件学士学位。目前，他在 Seneca 开源研究实验室（CDOT）担任助理研究员和技术主管，从事 WebGL 库的开发工作，其中包括 Processing. Js、C3DL 和 XB PointStream，并分别在 SIGGRAPH、MIT 和 Seneca 开源学术报告会展示了他的研究成果。

Giles Thomas 于 7 岁时接触了 ICL DRS 20，并开始了其编码生涯。自此开始，他的激情从未消退。12 岁时，他编写了个人的首个编程语言；14 岁时，他开发了首个操作系统。目前，他致力于改进一款名为 PythonAnywhere 的云计算系统。平时，他的言行多反映于博客上，其博客地址是 http://learningwebgl.com/。

致谢

本书的出版得益于以下人员的支持，他们是：我的同事 Jose，感谢你的爱和无限的耐心；我的家庭成员 Cecy、Fredy 和 Jonathan；我的良师益友 Terry Peters 博士和 Robert Bartha 博士，感谢你们的支持和鼓励；感谢我的朋友兼同事 Danielle Pace 和 Chris Russ，你们的正直、专业以及奉献精神给予了我极大的鼓舞，谢谢你们在本书编写时给予我的极大支持。

感谢我的合作作者Brandon Jones，他的出现使得WebGL世界变得更加精彩。Brandon参与了glMatrix库、第7章以及第10章的编写工作，并为本书的出版付出了辛勤的劳动。

在本书的编写过程中，审校人员功不可没，他们是 Dan Ginsburg、Giles Thomas、Andor Salga 和 Paul Brunt，感谢你们精彩的见解。

感谢辛勤的 PACKT 团队，以及为本书出版付出劳动的相关人士，他们是 Joel Goveya、Wilson D'souza、Maitreya Bhakal、Meeta Rajan、Azharuddin Sheikh、Manasi Poonthott am、Manali Mehta、Manmeet Singh Vasir、Archana Manjrekar 和 Duane Moraes。

最后，感谢我的妻子 Emily，爱犬 Cooper，感谢你们给予我极大的耐心；感谢 Zach，是你让我变得更加自信。

读者反馈和技术支持

欢迎读者对本书的建议或意见予以反馈，以进一步了解读者的阅读喜好。反馈意见对于我们来说十分重要，以便改进我们日后的工作。

对此，读者可向 feedback@packtpub.com 发送邮件，并以书名作为邮件标题。

若读者意欲查询出版信息，则可在 www.packtpub.com 网站的 SUGGEST A TITLE 表项中填写相关信息，或发送邮件至 suggest@packtpub.com。

若读者针对某项技术具有专家级的见解，抑或计划撰写书籍或完善某部著作的出版工作，则可阅读 www.packtpub.com/authors 中的 author guide 一栏。

目　录

第 1 章　WebGL 概述

2007 年，软件工程师 Vladimir Vukicevic 针对 HTML <canvas>——即后来的 canvas 3D 着手制定相关标准。2011 年 3 月，Vladimir Vukicevic 与 Kronos Group（OpenGL 幕后的一家非营利组织）联手推出了 WebGL 标准，该规范使得互联网浏览器可对图形处理单元（GPU）予以访问。

WebGL 源自 OpenGL ES 2.0（ES 是指嵌入式系统），即某些移动设备上的 OpenGL 规范，如 Apple 公司所研制的 iPhone 和 iPad。在随后的发展过程中，该规范逐渐脱离于最初的既定目标，即提供不同操作系统以及设备之间的可移植性。对于 3D Web 环境，如视频游戏、专业的可视化领域以及医学成像，基于 Web 的实时渲染理念则提供了更为广阔的天地。除此之外，考虑到 Web 浏览器的普及性，移动设备也可视为一类重要的应用平台，其中包括智能手机和平板电脑。若计划打造一款 Web 视频游戏、构建虚拟 3D 可视环境、实现数据可视化操作，或者是生成想象中的 3D 应用程序，则需要首先了解开发环境是否可满足当前的需求。

本章内容包含以下要点：

- 理解 Web 应用程序结构。
- 设置绘制区域（即 canvas）。
- 理解 WebGL 的状态机机制。
- 调整 WebGL 变量以修改当前场景环境。
- 加载并检测全功能场景环境。

1.1　系 统 需 求

WebGL 可视为一类 3D Web 图形 API，因而无须执行安装步骤，但应提供以下一种互联网浏览器。

- Firefox 4.0 浏览器（或以上版本）。
- Google Chrome 11 浏览器（或以上版本）。
- Safari 浏览器（OSX 10.6 或以上版本）。默认状态下，Safari 禁用 WebGL，读者可查看 Developer 菜单并选取 WebGL 选项。

- Opera 12 浏览器（或以上版本）。

为了获得 WebGL 所支持的互联网 Web 浏览器更新列表，读者可访问 Khronos Group 网站，对应网址为 http://www.khronos.org/webgl/wiki/Getting_a_WebGL_Implementation。除此之外，读者还应在自己的机器设备上配有图形卡。同时，还可进一步访问 http://get.webgl.org/以获取 WebGL 所支持的环境配置。

1.2　WebGL 提供的渲染类型

WebGL 定义为一个 3D 图形库，并以标准、高效的方式支持互联网浏览器的 3D 场景渲染操作。根据维基百科中所描述的内容，渲染操作可视为源自模型的图像生成过程，并通过计算机程序予以实现。由于该过程可表示为计算机处理行为，因而存在多种方式生成此类图像。

差别一体现于是否需要使用特定的图形硬件。对于软件渲染，3D 场景所需计算均在 CPU 中执行；相应地，硬件渲染则是指图形处理单元（GPU）以实时方式执行 3D 图形计算。从技术视角来看，由于硬件负责某些专属操作，因而与软件渲染相比，其行为更加高效。鉴于独立于硬件支持，因而软件渲染方案应用范围较大。

差别二是指渲染处理过程出现于本地端或远程端。若渲染图像过于复杂，则渲染器通常位于远程一端，如 3D 动画电影，复杂图像通常由配以大量硬件资源的服务器负责渲染，即服务器渲染。相反，若渲染过程出现于本地，则该过程称作客户端渲染。

WebGL 采用了客户端渲染方案，3D 场景素材通过服务器下载，但图像处理过程依赖于客户端的图形硬件并在本地予以执行。

与其他技术相比（如 Java 3D、Flash 以及 Unity Web Player Plugin），WebGL 具有以下优点。

- JavaScript 编程机制。对于 Web 开发人员以及互联网 Web 浏览器，JavaScript 可视为相对自然的程序设计语言，并支持 DOM 访问。除此之外，JavaScript 还可方便地在各元素间进行通信。由于 WebGL 在 JavaScript 中进行编程，因而可方便地与其他 JavaScript 库以及 HTML 5 技术进行整合。
- 自动内存管理。在 OpenGL 以及其他相关技术中，读者需手工分配或释放内存空间，而 WebGL 无须执行此项操作。WebGL 遵循变量作用域规则，并在必要时自动释放内存空间。这简化了程序设计过程，并降低了代码的书写量，进而使程序易于理解。
- 性能问题。WebGL 程序的性能可与独立的应用程序相媲美，这一切均归功于

WebGL 的本地图形硬件访问能力。当前，多数 3D Web 渲染技术均采用软件渲染方案。

- 零编译。由于 WebGL 于 JavaScript 中实现，因而在运行前无须进行编译，并可实现即时调整。据此，读者可尝试观察此类变化对 3D Web 应用程序所带来的影响。尽管如此，当讨论着色器程序时，编译过程依然不可或缺，但该过程出现于图形硬件中，而非浏览器。

1.3　WebGL 应用程序结构

类似于其他 3D 图形库，WebGL 同样需要特定组件的支持，进而构造 3D 场景环境。本书第 1~4 章将讨论此类基本要素，并于随后探讨 3D 场景之外的其他元素，如颜色和纹理。最后，本书还将对某些高级话题予以分析。

其相关组件包括如下内容。

- canvas：作为一类容器，承载即将渲染的场景，此类场景多由 HTML 5 元素构成，并可通过文档对象模型（DOM）进行访问。
- 对象：部分场景还采用了 3D 实体对象，且由三角形构成。第 2 章将考察 WebGL 针对三角形的处理方式。其中，WebGL 通过缓冲区机制存储多边形数据，并于随后渲染场景中的对象。
- 光照：若缺乏光照的支持，3D 场景将难以呈现其应有外观。第 3 章将对此予以阐述，其中，WebGL 通过着色器对场景中的光照进行建模，3D 对象根据物理定律反射、吸收光线。除此之外，第 3 章还将讨论 WebGL 中的不同光照模型，进而实现场景对象的可视化操作。
- 相机：canvas 通常视为 3D 场景世界的视口，并以此对当前场景进行查看。第 4 章将讨论各类矩阵操作，进而生成视见透视效果。另外，第 4 章还将进一步分析此类操作基于相机的建模方式。

本章将讨论上述第 1 项内容 canvas，包括 canvas 的生成方式以及 WebGL 上下文环境的构建方式。

1.4　HTML 5 canvas 的生成方式

下面讨论 Web 页的生成过程，并向其添加 HTML 5 canvas。此处，canvas 可视为 Web 页中渲染 3D 场景时的矩形元素。