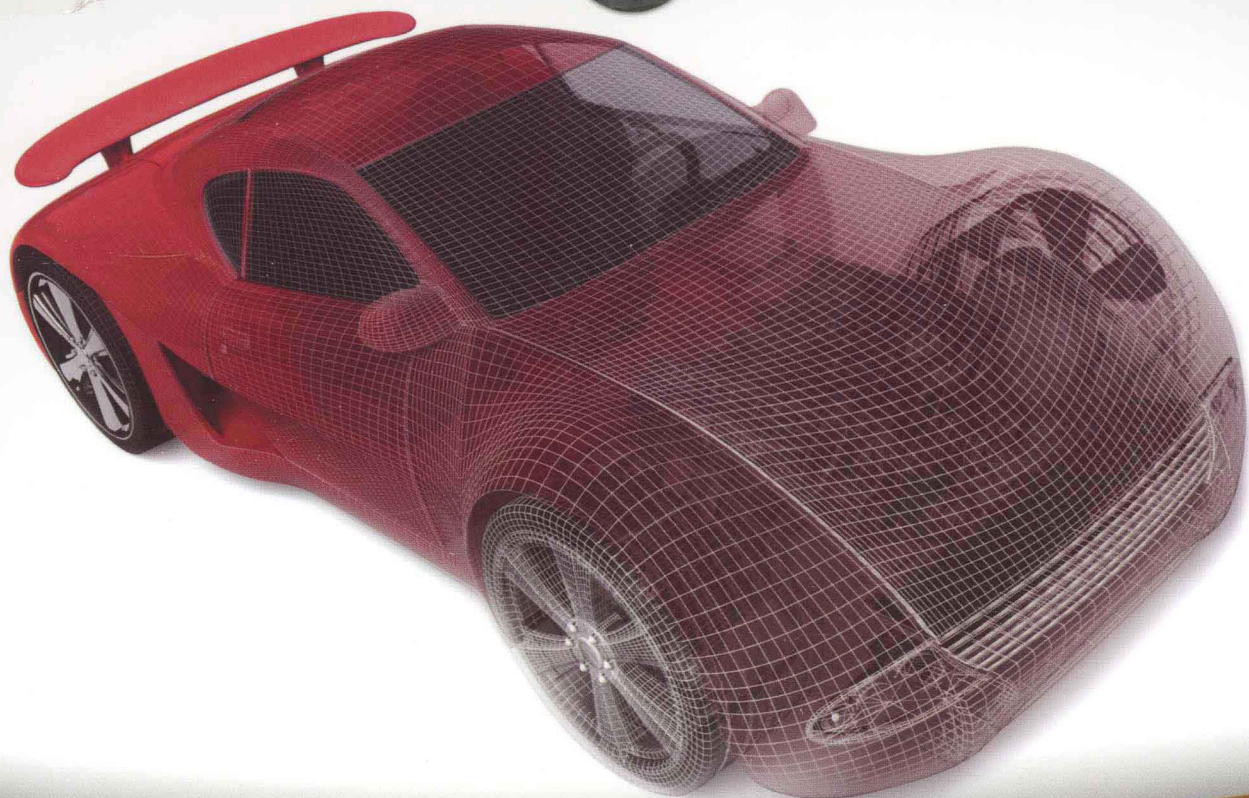




Professional WebGL Programming: Developing 3D
Graphics for the Web

WebGL高级编程

——开发Web 3D图形

[美] Andreas Anyuru 著
吴文国 译



清华大学出版社

WebGL 高级编程

——开发 Web 3D 图形

[美] Andreas Anyuru 著
吴文国 译

清华大学出版社

北 京

Andreas Anyuru

Professional WebGL Programming: Developing 3D Graphics for the Web

EISBN: 978-1-119-96886-3

Copyright © 2012 by John Wiley & Sons, Inc.

All Rights Reserved. This translation published under license.

本书中文简体字版由 Wiley Publishing, Inc. 授权清华大学出版社出版。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。

北京市版权局著作权合同登记号 图字: 01-2012-8055

本书封面贴有 Wiley 公司防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

WebGL 高级编程——开发 Web 3D 图形/(美)阿尤鲁(Anyuru, A.) 著; 吴文国 译. —北京: 清华大学出版社, 2013.6

书名原文: Professional WebGL Programming: Developing 3D Graphics for the Web

ISBN 978-7-302-32183-5

I. ①W… II. ①阿… ②吴… III. ①网页制作工具—程序设计 IV. ①TP393.092

中国版本图书馆 CIP 数据核字(2013)第 083362 号

责任编辑: 王 军 于 平

装帧设计: 牛静敏

责任校对: 邱晓玉

责任印制: 刘海龙

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: 清华大学印刷厂

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 19.5 字 数: 475 千字

版 次: 2013 年 6 月第 1 版 印 次: 2013 年 6 月第 1 次印刷

印 数: 1~3000

定 价: 49.00 元

作者简介

Andreas Anyuru 精通包括 WebGL 在内的许多 Web 技术，他在移动设备 Web 浏览器集成和优化方面曾有多年的工作经历。Andreas 是 V8 JavaScript 引擎的贡献者之一，后者用在 Google 台式浏览器 Chrome 和 Android 手机中。他设计了几个新教程，并在 Lund 大学的信息和通信工程学院任教。

Andreas 是 ST-爱立信公司的高级技术人员，主要研究领域是 Web 技术。主要工作职责是确保现有的和即将推出的 Web 技术以最优的方式应用于 ST-爱立信基于 Linux 的移动平台上。

技术编辑简介

Paul Brunt 在 HTML5 刚推向市场时就用 JavaScript 开发了几个游戏和应用程序，这些游戏和应用程序广泛应用了 SVG、画布和新一代 JavaScript 引擎等技术。他的工作还包括一个名为“Berts Breakdown”的概念验证平台的游戏演示程序。他对计算机艺术怀有浓厚的兴趣，在 Blender 和实时图形方面具有渊博的知识，WebGL 的发布激发了他开发 GLGE 的欲望。2009 年，当时 WebGL 还处于婴儿期，他就开始开发 GLGE，然后以锐不可挡之势转向在线游戏开发。他对其他 WebGL 架构和项目也有贡献；另外他于 2011 年将 jiglib 物理库移植到 JavaScript 中，并首次在浏览器里演示 3D 物理。

致 谢

首先要感谢提出 WebGL 规范的 Khronos WebGL 工作小组、浏览器供应商以及实现了 WebGL 规范的开源项目组。没有你们的工作，就没有本书的问世。此外，感谢 WebGL 社区的所有开发人员，你们的讨论、经历和示例使 WebGL 成为一项技术。

感谢 Wiley 出版社的员工，特别是 Jeff Riley、Sara Shlaer、Ellie Scot 和 Chris Webb，感谢你们给我工作上的支持！

感谢 ST-爱立信公司的所有同仁，有了你们，ST-爱立信才成为一个工作愉快的地方；特别感谢 Kent Olsson、Christian Bejram、Marco Cornero、Göran Roth 和 Patrik Aberg，感谢你们给我的信任、支持和指导。

感谢我的家人对我工作的支持和理解，最重要的是，我要感谢我的太太 Jessica。Jessica，你是我的最爱，你是我的一切！

前言

现在用户将很大一部分时间花在自己的计算机、平板电脑和智能手机上冲浪。用户整天忙于编写文档、发送和接收电子邮件、聊天、观看视频、欣赏音乐、玩游戏或购物。Web 浏览器已成为大多数设备上最重要的应用程序。结果是，Web 新技术的发展日新月异。现在 Web 浏览器变得越来越快、越来越稳定、越来越安全，它能够处理每天出现的新技术。

这是一个令 Web 开发人员兴奋的时代。如果想开发一个应用程序，并且希望将它发布给众多用户使用，则 Web 无疑是首选的平台。尽管 Web 技术在最近几年里已快速发展，但是与其他原生应用程序(Native Application)相比，它有一个缺点，即 Web 技术无法创建游戏和其他应用程序所需要的实时 3D 图形。WebGL 的出现改变了这一切。

WebGL 以前所未有的方式允许用户硬件加速 2D 和 3D 图形。用户既可以应用 HTML、CSS 和 JavaScript 提供的全部优点，同时又可以应用功能强大的图形处理单元(Graphics Processor Unit, GPU)提供的好处。本书先向用户介绍开始开发基于 WebGL 的 2D 或 3D 图形应用程序所需的全部基础知识和准备工作。

本书读者对象

本书的主要读者是已对 HTML、CSS 和 JavaScript 有一个基本了解，但是想深入学习如何用 WebGL 为自己的 Web 应用程序或网页创建硬件加速的 2D 或 3D 图形程序的用户。

能够从本书获益的第二类读者包括那些曾经开发过台式 3D API(如 OpenGL 或 Direct3D 等)，但是想在 Web 上把这些知识应用于 WebGL 程序开发的开发人员。

对于学习 3D 图形课程以及想把 WebGL 作为创建原型和测试程序的一个简单工具的学生来说，本书也是十分有用的。因为你们只需要一个支持 WebGL 的 Web 浏览器和一个可以用来编写应用程序的文本编辑器，编写 WebGL 程序的门槛远比其他 3D 图形 API 低(如台式 OpenGL 或 Direct3D)，后者需要一个完整的工具链。

本书主要内容

本书向读者介绍如何开发基于 WebGL 的 Web 应用程序。虽然 WebGL API 可用来硬件加速 2D 图形和 3D 图形，但是它的主要作用是用来创建 3D 图形。3D 图形 API 的一些图书只介绍 API 本身，并没有对 3D 图形或如何使用 API 进行较多的介绍。本书不要求读者

具备任何 3D 图形的理论基础。希望读者通过本书的学习能够掌握 3D 图形基础知识，以及学会用 WebGL API 开发 Web 应用程序。

此外，本书还介绍线性代数的部分基础知识，这有助于读者深入理解 3D 图形和 WebGL 底层的运行机制。掌握了线性代数的基本知识，读者就可以把重点放在线性代数中 3D 图形重要的部分。读者不需要去阅读厚达几百页的通用线性代数教材，这些图书通常以通用和抽象的方式介绍每个专题。如果读者属于只想很快开始编写代码的一类用户，不需要阅读线性代数一节的全部内容(主要是第 1 章的部分内容)。读者可以跳过这部分内容。若后来发现某些问题与线性代数有关，则可以回过头来再仔细阅读相关内容。

必须向读者指出，本书并没有介绍 WebGL 的全部方法。要想得到 WebGL API 的完整参考手册，则至少要拥有一本 Khronos 发布在 www.khronos.org/registry/webgl/specs/latest/ 上的有关最新 WebGL 规范的图书。

本书的组织结构

本书采用一种有利于大多数读者阅读的逻辑顺序。然而，由于读者有不同的背景，因此他们可以根据自己的需要选择最适合自己的顺序阅读本书。下面的概述可能会有助于读者安排好读书计划。

第 1 章介绍理论知识。向读者介绍 WebGL 的历史背景，并把它与其他几个图形技术进行比较。这一章还向读者介绍 WebGL 图形流水线结构，同时也对图形硬件做概括性介绍。此外，本章还介绍线性代数的部分基础知识，这些内容有助于读者更好地理解 3D 图形和 WebGL 的底层工作机制。如果读者认为自己对本章介绍的线性代数有足够的知识，则可以跳过这一部分内容，直接阅读第 2 章。

第 2 章向读者介绍很多实用知识。介绍如何建立第一个完整的 WebGL 应用程序和如何编写一个非常简单的顶点着色器和片段着色器。为了尽可能方便读者后面的学习，本章还介绍几个有用的开发和调试工具。此外，还会介绍如何排除 WebGL 应用程序中的错误。

第 3 章向读者详细介绍 WebGL 绘制的各个不同选项。读者要学习在绘图中可以使用的各个 WebGL 方法和图元。

第 4 章介绍 3 个小型的 JavaScript 图形库，它们常用来执行 WebGL 应用程序中的矢量和矩阵运算。此外，读者还要学习如何用变换运算确定对象在 3D 空间里的位置和朝向。这一章非常重要，如果读者以前没有 3D 图形方面的经历，则要在这一章多花一些时间。

第 5 章介绍 WebGL 的纹理贴图。纹理贴图是使 3D 对象具有真实感的一个重要步骤。此外，本章还介绍如何通过 WebGL 中丢失上下文的处理使自己的程序更加健壮。在设计真实应用程序之前一定要仔细阅读这部分的内容，因为它们对健壮性有很高的要求。

第 6 章介绍如何使用动画技术创建 WebGL 场景的动画效果。此外在这一章读者还要学习 JavaScript 的事件处理模式，以及如何利用键盘事件和鼠标事件控制 WebGL 场景。

第 7 章介绍 WebGL 的光照处理，这是一个令人兴奋的主题。读者学习各种光照模型和如何用这些光照模型编写顶点着色器和片段着色器。光照是使 WebGL 场景具有比较真实效果的一项重要技术。

第 8 章包含一些指南、提示和使用技巧，读者利用这些技术尽可能提高自己的 WebGL 应用程序的性能。此外，这一章还分析 WebGL 应用程序的性能问题，如何找到瓶颈位置，如何消除这些瓶颈。该章也会介绍 WebGL 的底层工作机制。这一章还向读者介绍关键软件成分和关键硬件成分。最后本章将介绍移动行业里的一个示例。

阅读本书之前的准备工作

为了运行本书的例子，读者需要一个支持 WebGL 的浏览器。如果读者的系统还没有这样一个浏览器，可以免费下载一个。想知道当前支持 WebGL 的 Web 浏览器列表，请访问 www.khronos.org/webgl/wiki/。在编写本书的时候，以下浏览器支持 WebGL：

- Apple Safari
- Google Chrome
- Mozilla Firefox
- Opera

为了编写自己的 WebGL 应用程序，要使用自己喜欢的文本编辑器。本书还介绍程序调试和程序故障排除的几个十分有用的工具，它们是 Chrome 开发人员工具(Chrome Developer Tools)、Firebug 和 WebGL Inspector。它们都属于开源代码工具，可以免费下载。

源代码

在读者学习本书中的示例时，可以手动输入所有代码，也可以使用本书附带的源代码文件。本书使用的所有源代码都可以从本书合作站点 <http://www.wrox.com/>或 <http://www.tupwk.com.cn/download> 上下载。登录到站点 <http://www.wrox.com/>，使用 Search 工具或使用书名列表就可以找到本书。接着单击 Download Code 链接，就可以获得所有的源代码。既可以选择下载一个大的包含本书所有代码的 ZIP 文件，也可以只下载某个章节中的代码。



由于许多图书的标题都很类似，因此按 ISBN 搜索是最简单的，本书英文版的 ISBN 是 978-1-119-96886-3。

在下载代码后，只需用解压缩软件对它进行解压缩即可。另外，也可以进入 <http://www.wrox.com/dynamic/books/download.aspx> 上的 Wrox 代码下载主页，查看本书和其他 Wrox 图书的所有代码。记住，可以使用书中列出的程序清单的编号容易地找到所要寻找的代码，

如“程序清单 0-1”。

当为大多数可下载的源代码文件命名时，我们会使用这些清单中的数值。对于那些很少的没有用它自己的清单数值命名的程序清单，它们都与文件名匹配，所以很容易就可以在下下载的源代码文件中找到它们。

勘误表

尽管我们已经尽了各种努力来保证文章或代码中不出现错误，但是错误总是难免的，如果您在本书中找到了错误，例如拼写错误或代码错误，请告诉我们，我们将非常感激。通过勘误表，可以让其他读者避免受挫，当然，这还有助于提供更高质量的信息。

要在网站上找到本书英文版的勘误表，可以登录 <http://www.wrox.com>，通过 Search 工具或书名列表查找本书，然后在本书的细目页面上，单击 Book Errata 链接。在这个页面上可以查看到 Wrox 编辑已提交和粘贴的所有勘误项。完整的图书列表还包括每本书的勘误表，网址是 www.wrox.com/misc-pages/booklist.shtml。

如果您发现的错误在我们的勘误表里还没有出现的话，请登录 www.wrox.com/contact/techsupport.shtml 并完成那里的表格，把您发现的错误发送给我们。我们会检查您的反馈信息，如果正确，我们将在本书的勘误表页面张贴该错误消息，并在本书的后续版本加以修订。

p2p.wrox.com

要与作者和同行讨论，请加入 p2p.wrox.com 上的 P2P 论坛。这个论坛是一个基于 Web 的系统，便于您张贴与 Wrox 图书相关的消息和相关技术，与其他读者和技术用户交流心得。该论坛提供了订阅功能，当论坛上有新的消息时，它可以给您传送感兴趣的论题。Wrox 作者、编辑和其他业界专家和读者都会到这个论坛上来探讨问题。

在 <http://p2p.wrox.com> 上，有许多不同的论坛，它们不仅有助于阅读本书，还有助于开发自己的应用程序。要加入论坛，可以遵循下面的步骤：

- (1) 进入 p2p.wrox.com，单击 Register 链接。
- (2) 阅读使用协议，并单击 Agree 按钮。
- (3) 填写加入该论坛所需要的信息和自己希望提供的其他可选信息，单击 Submit 按钮。您会收到一封电子邮件，其中的信息描述了如何验证账户，完成加入过程。



不加入 P2P 也可以阅读论坛上的消息，但要张贴自己的消息，就必须先加入该论坛。

加入论坛后，就可以张贴新消息，响应其他用户张贴的消息。可以随时在 Web 上阅读消息。如果要让该网站给自己发送特定论坛中的消息，可以单击论坛列表中该论坛名旁边的 **Subscribe to this Forum** 图标。

要想了解更多的有关论坛软件的工作情况，以及 P2P 和 Wrox 图书的许多常见问题的解答，就一定要阅读 FAQ，只需在任意 P2P 页面上单击 FAQ 链接即可。

目 录

第 1 章 WebGL 简介	1
1.1 WebGL 基础	1
1.2 浏览器 3D 图形吸引人的原因	2
1.3 设计一个图形 API	3
1.3.1 即时模式 API	3
1.3.2 保留模式 API	3
1.4 图形硬件简介	4
1.4.1 GPU	4
1.4.2 帧缓存	5
1.4.3 纹理存储器	6
1.4.4 视频控制器	6
1.5 WebGL 图形流水线	6
1.5.1 顶点着色器	8
1.5.2 图元装配	11
1.5.3 光栅化	12
1.5.4 片段着色器	12
1.5.5 逐片段操作	15
1.6 WebGL 与其他图形技术的比较	16
1.6.1 OpenGL	16
1.6.2 OpenGL ES 2.0	18
1.6.3 Direct3D	19
1.6.4 HTML5 画布	21
1.6.5 可缩放矢量图形	25
1.6.6 VRML 与 X3D	26
1.7 线性代数简介	27
1.7.1 坐标系	27
1.7.2 点与顶点	27
1.7.3 矢量	28
1.7.4 矢量的点积或标积	29

1.7.5 叉积	30
1.7.6 齐次坐标	31
1.7.7 矩阵	31
1.7.8 仿射变换	34
1.8 小结	39
第 2 章 创建基本的 WebGL 示例	41
2.1 绘制三角形	41
2.1.1 创建 WebGL 上下文	45
2.1.2 创建顶点着色器和片段着色器	47
2.1.3 编译着色器	48
2.1.4 创建程序对象和链接着色器	48
2.1.5 建立缓冲	50
2.1.6 绘制场景	51
2.2 了解 WebGL 编码风格	52
2.3 调试 WebGL 应用程序	53
2.3.1 使用 Chrome 开发人员工具	53
2.3.2 Firebug 的使用	59
2.3.3 WebGL 的错误处理与错误代码	61
2.3.4 WebGL Inspector	64
2.3.5 WebGL 的故障排除	70
2.4 用 DOM API 载入着色器	71
2.5 更高级的综合示例	73
2.6 小结	77
第 3 章 绘制	79
3.1 使用 WebGL 绘制图元和绘图方法	79

3.1.1	图元	80	4.4.3	将变换矩阵上传给 GPU 中的顶点着色器	148
3.1.2	顶点组绕顺序的重要性	84	4.4.4	调用绘图方法	149
3.1.3	WebGL 的绘图方法	85	4.5	理解变换顺序的重要性	150
3.2	类型化数组	91	4.5.1	使用一个固定的全局的 坐标系	150
3.2.1	缓冲与视图	91	4.5.2	使用移动的局部的 坐标系	153
3.2.2	WebGL 支持的视图类型	92	4.5.3	变换矩阵的入栈和出栈 操作	154
3.3	探讨不同的绘图方法	93	4.6	一个完整的示例：绘制几个 变换后的对象	157
3.3.1	gl.drawArrays()和 gl.TRIANGLES	94	4.6.1	使用 WebGL 代码创建 立方体	159
3.3.2	gl.drawArrays()方法和 gl.TRIANGLE_STRIP 图元	96	4.6.2	视图变换和模型变换的 组织	161
3.3.3	gl.drawElements()方法和 gl.TRIANGLES 图元	98	4.7	小结	162
3.3.4	gl.drawElements()方法和 gl.TRIANGLE_STRIP 图元	100	第 5 章	纹理贴图	163
3.3.5	总结比较	102	5.1	理解丢失上下文	164
3.3.6	前期变换顶点缓存和 后期变换顶点缓存	102	5.1.1	理解解决丢失上下文问题 所需要的设置	164
3.4	为提高性能交叉存放顶点 数据	104	5.1.2	处理丢失上下文问题时 需要考虑的几个因素	166
3.5	使用顶点数组或常量顶点 数据	113	5.2	2D 纹理与立方映射纹理	169
3.6	总结本章的最后一个示例	114	5.3	载入纹理	170
3.7	小结	124	5.3.1	创建 WebGLTexture 对象	170
第 4 章	小型 JavaScript 库与变换	127	5.3.2	绑定纹理	171
4.1	JavaScript 中矩阵和向量的 操作	127	5.3.3	载入图像数据	171
4.1.1	Sylvester 库	128	5.3.4	将纹理上传到 GPU	173
4.1.2	WebGL-mjs 库	132	5.3.5	定义纹理参数	174
4.1.3	glMatrix 库	135	5.3.6	理解载入纹理的完整 过程	175
4.2	变换运算	139	5.3.7	创建一个纹理对象并载入 纹理数据	177
4.3	理解完整的变换流水线	145	5.4	定义纹理坐标	178
4.4	变换的实践	146	5.5	着色器中的纹理处理	180
4.4.1	为对象坐标设置缓冲	147			
4.4.2	用 JavaScript 创建变换 矩阵并上传给着色器	148			

5.6 处理纹理过滤.....183	6.2 用户交互事件的处理.....213
5.6.1 纹理伸展.....184	6.2.1 DOM Level 0 基本事件 处理.....214
5.6.2 纹理收缩.....185	6.2.2 DOM Level 2——高级 事件处理方法.....215
5.6.3 Mip 映射纹理.....186	6.2.3 键盘输入.....217
5.7 理解纹理坐标包装.....188	6.2.4 鼠标输入.....221
5.7.1 应用 gl.REPEAT 包装 模式.....188	6.3 综合应用新知识.....223
5.7.2 应用 gl.MIRRORED_ REPEAT 包装模式.....190	6.4 小结.....229
5.7.3 应用 gl.CLAMP_TO_ EDGE 包装模式.....191	第 7 章 光照.....231
5.8 一个完整的应用纹理示例.....191	7.1 光源.....231
5.9 获得用作纹理的图像.....194	7.2 局部光照模型的工作原理.....232
5.9.1 下载免费纹理.....194	7.3 Phong 反射模型.....232
5.9.2 用自己拍摄的照片生成 纹理.....194	7.3.1 环境反射.....233
5.9.3 绘制图像.....195	7.3.2 漫反射.....234
5.9.4 购买纹理.....195	7.3.3 镜面反射.....236
5.10 同域策略与跨域资源共享.....195	7.3.4 Phong 反射模型的完整 公式和着色器.....239
5.10.1 同域策略应用于一般的 图像.....196	7.3.5 光照效果与纹理相结合.....243
5.10.2 同域策略应用于纹理.....197	7.4 WebGL 光照中需要的 JavaScript 代码.....246
5.10.3 跨域资源共享.....199	7.4.1 为顶点法线设置缓存.....247
5.11 小结.....200	7.4.2 计算法线矩阵并上传给 着色器.....249
第 6 章 动画与用户输入.....203	7.4.3 将光照信息上传给 着色器.....250
6.1 创建动画场景.....203	7.5 将不同的插值方法用于 着色.....250
6.1.1 setInterval()和 setTimeout()的 使用.....205	7.5.1 平面着色.....251
6.1.2 使用 requestAnimationFrame() 函数.....206	7.5.2 Gouraud 着色.....252
6.1.3 帧频不同引起的运动 补偿.....209	7.5.3 Phong 着色.....253
6.1.4 创建 FPS 计数器测量 动画的平稳性.....210	7.6 矢量必须归一化.....256
6.1.5 用 FPS 作为测量值的 缺点.....212	7.6.1 顶点着色器中的矢量 归一化.....257
	7.6.2 片段着色器的矢量 归一化.....257

7.7 应用不同类型的光源.....258	8.2.6 改善像素受限的 WebGL.....285
7.7.1 平行光.....258	8.3 深入分析融合.....286
7.7.2 点光源.....259	8.3.1 融合简介.....286
7.7.3 聚光源.....259	8.3.2 设置融合函数.....287
7.8 光强衰减.....262	8.3.3 绘制顺序与深度缓冲区.....290
7.9 光照映射.....265	8.3.4 绘制包含不透明对象和半透明对象的场景.....290
7.10 小结.....267	8.3.5 修改融合公式中的默认运算符.....291
第 8 章 WebGL 性能优化.....269	8.3.6 使用预乘 alpha 值.....292
8.1 WebGL 底层工作机制.....269	8.4 深入讨论 WebGL.....292
8.1.1 支持 WebGL 的硬件.....270	8.4.1 使用 WebGL 框架.....293
8.1.2 关键的软件组成.....271	8.4.2 发布到 Google Chrome Web Store.....293
8.2 WebGL 性能优化.....274	8.4.3 使用额外资源.....293
8.2.1 避免初学者的典型错误.....274	8.5 小结.....294
8.2.2 确定瓶颈位置.....275	
8.2.3 有关性能的一般性建议.....279	
8.2.4 改善 CPU 受限的 WebGL 应用程序性能的建议.....282	
8.2.5 改善顶点受限的 WebGL 应用程序性能的建议.....283	

第 1 章

WebGL 简介

本章主要内容：

- WebGL 基础
- 浏览器 3D 图形吸引人的原因
- 如何使用即时模式 API
- 图形硬件初步
- WebGL 图形流水线
- WebGL 与其他图形技术的比较
- 学习 WebGL 必备的线性代数知识

本章介绍 WebGL、它的一些重要理论以及 3D 图形的一般性知识。通过本章的学习，你可以掌握什么是 WebGL，并对 WebGL 使用的图形流水线有一个基本的理解。

此外，为了使你对 WebGL 有一个更全面的了解，本章还介绍了其他几个相关的图形标准，并且把 WebGL 与这些图形技术进行比较。

最后，本章复习了线性代数的一些基本理论，如果你确实想更深层掌握 WebGL 技术，这些知识是必不可少的。

1.1 WebGL 基础

WebGL 是一个用来在 Web 上生成三维图形效果的应用编程接口。它是以 OpenGL ES 2.0 为基础的。与 OpenGL 一样，它也提供绘制(Rendering(API))功能，但是它应用在 HTML 和 JavaScript 上下文中。WebGL 绘制的曲面本质上是 HTML5 的画布，后者最早是由 Apple 公司引入到 WebKit 开源浏览器引擎中的。引入 HTML5 画布的理由是为了能够在 Dashboard 小部件(Widget)等应用程序中和 Apple Mac OS X 操作系统的 Safari 浏览器中绘制二维图形。

以此画布原理为基础, Mozilla 公司的 Vladimir Vukicevic 开始试验画布的三维图形效果。他称最初的模型为 3D 画布(Canvas 3D)。2009 年, Khronos Group 开始组建新的 WebGL 工作组。现在这个工作组由 Apple、Google、Mozilla 和 Opera 等几个主要浏览器的供应商组成。Khronos Group 是一个非营利的行业同盟, 制订了许多开放标准和无版权费的 API。它成立于 2000 年 1 月。除此之外, 该小组还开发了许多其他 API 并提出了其他图形技术, 如用于嵌入式设备的 OpenGL ES, 用于并行程序设计的 OpenCL, 用于矢量图形低层加速的 OpenVG, 用于加速多媒体部件的 OpenMAX。2006 年, Khronos Group 开始接管并升级了 OpenGL。后者是台式机的 3D 图形 API。

WebGL 1.0 规范最终于 2011 年 3 月确定下来, 并且在 Google Chrome、Mozilla Firefox 等浏览器中以及在 Safari 和 Opera 等的开发过程中(在本书编写时)实现对它的支持。



有关不同浏览器对 WebGL 支持的最新消息, 请访问 www.khronos.org/webgl/。

1.2 浏览器 3D 图形吸引人的原因

在万维网的早期, 网页的内容只包含静态的文本文档, 它们由 Web 浏览器读取和显示。近几年中, Web 技术已得到巨大的发展, 现在许多网站实际上是全功能的应用程序。它们支持服务器与客户端之间的双向通信, 允许用户注册和登录, 而且 Web 应用程序提供了丰富的用户界面, 如图形、声音和视频。

Web 应用程序的快速发展促使它们成为原生应用程序的一个强有力的竞争者。Web 应用程序包括以下几个优点:

- 可以廉价迅速地传播给大量的用户。用户只需要一个兼容的 Web 浏览器。
- 容易维护。当我们发现应用程序有一个 bug, 或者当我们想给应用程序添加有利于用户的新功能时, 只需要升级 Web 服务器上的应用程序, 用户就可以立刻使用升级后的应用程序。
- 由于应用程序是在 Web 浏览器中执行的, 因此相对比较容易得到跨平台的支持(如得到 Windows、Mac OS、Linux 等操作系统的支持)。至少在理论上是如此。

然而, 与原生应用程序相比, Web 应用程序也存在有一些局限性。其中之一是 Web 应用程序的用户界面无法做到原生应用程序那样丰富。自从引入了 HTML5 画布标记以来, 这种情形已发生很大的变化。有了它, 我们可以为 Web 应用程序创建真正高级的 2D 图形。但是早期的 HTML5 画布标记只允许定义 2D 上下文, 并不支持 3D 图形。

但是, 自从有了 WebGL, 开发人员就可以在浏览器内部实现 3D 图形的硬件加速, 就可以创建 3D 游戏或者其他高级的 3D 图形应用程序。另一方面, 它具有 Web 应用程序的全部优点。WebGL 具有以下吸引人的特性:

- WebGL 是一个开放的标准, 任何人都可以使用, 不需要支付任何版权费。

- WebGL 利用图形硬件加速图形绘制，这意味着它的速度确实很快。
- WebGL 可以在支持它的本地浏览器上运行，不需要任何插件。
- 由于 WebGL 是以 OpenGL ES 2.0 为基础的，因此对于具有 OpenGL ES 2.0 编程经验的开发人员而言，甚至对于熟悉台式机 OpenGL 开发的人们而言，它是很容易学习的。

WebGL 标准也为学生和其他人员提供了一个快速学习和测试 3D 图形的方法。不像大多数其他 3D API 那样，需要下载和安装一个工具链。用户只需要一个编写代码的文本编辑器，为了查看生成结果，只需要把这个文件载入到支持 WebGL 的浏览器中。

1.3 设计一个图形 API

设计一个图形 API，基本上采用以下两种不同的模式：

- 即时模式 API
- 保留模式 API

WebGL 属于即时模式 API。

1.3.1 即时模式 API

对于即时模式 API(immediate-mode API)，每一帧的场景，不管是否已发生变化，都需要重新绘制。提供 API 的图形库并没有保存需要绘制场景的内部模型，但是应用程序需要在内存中用自己的方法表示场景。这种设计模式大大提高了应用程序的灵活性和控制能力。但是，它需要应用程序执行更多的操作，如跟踪场景的模型、执行初始化和清除操作。图 1-1 用简单的结构说明了即时模式 API 的绘制过程。

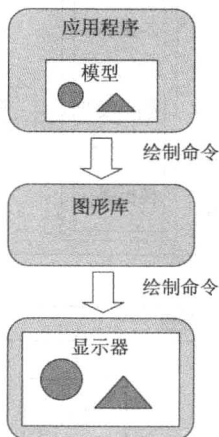


图 1-1 即时模式 API 的工作流程

1.3.2 保留模式 API

提供保留模式 API(retained-mode API)的图形库包含一个内部模型或场景图形。它包含