

<<WebGL高级编程>>

图书基本信息

书名：<<WebGL高级编程>>

13位ISBN编号：9787302321835

10位ISBN编号：7302321833

出版时间：2013-6

出版时间：清华大学出版社

作者：Andreas Anyuru

译者：吴文国

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

前言

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，不需要阅读线性代码一节的全部内容（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。

<<WebGL高级编程>>

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。

在开始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/downpage>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。

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。

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。

要了解更多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。

作者简介

作者：（美国）阿尤鲁（Anyuru, A.）译者：吴文国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V8JavaScript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。

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。

书籍目录

目录第1章 WebGL简介11.1 WebGL基础11.2 浏览器3D图形吸引人的原因21.3 设计一个图形API31.3.1 即时模式API31.3.2 保留模式API31.4 图形硬件简介41.4.1 GPU41.4.2 帧缓存51.4.3 纹理存储器61.4.4 视频控制器61.5 WebGL图形流水线61.5.1 顶点着色器81.5.2 图元装配111.5.3 光栅化121.5.4 片段着色器121.5.5 逐片段操作151.6 WebGL与其他图形技术的比较161.6.1 OpenGL161.6.2 OpenGL ES2 0181.6.3 Direct3D191.6.4 HTML5画布211.6.5 可缩放矢量图形251.6.6 VRML与X3D261.7 线性代数简介271.7.1 坐标系271.7.2 点与顶点271.7.3 矢量281.7.4 矢量的点积或标积291.7.5 叉积301.7.6 齐次坐标311.7.7 矩阵311.7.8 仿射变换341.8 小结39第2章 创建基本的WebGL示例412.1 绘制三角形412.1.1 创建WebGL上下文452.1.2 创建顶点着色器和片段着色器472.1.3 编译着色器482.1.4 创建程序对象和链接着色器482.1.5 建立缓冲502.1.6 绘制场景512.2 了解WebGL编码风格522.3 调试WebGL应用程序532.3.1 使用Chrome开发人员工具532.3.2 Firebug的使用592.3.3 WebGL的错误处理与错误代码612.3.4 WebGL Inspector642.3.5 WebGL的故障排除702.4 用DOM API载入着色器712.5 更高级的综合示例732.6 小结77第3章 绘制793.1 使用WebGL绘制图元和绘图方法793.1.1 图元803.1.2 顶点组绕顺序的重要性843.1.3 WebGL的绘图方法853.2 类型化数组913.2.1 缓冲与视图913.2.2 WebGL支持的视图类型923.3 探讨不同的绘图方法933.3.1 gl.drawArrays()和gl.TRIANGLES943.3.2 gl.drawArrays()方法和gl.TRIANGLE_STRIP图元963.3.3 gl.drawElements()方法和gl.TRIANGLES图元983.3.4 gl.drawElements()方法和gl.TRIANGLE_STRIP图元1003.3.5 总结比较1023.3.6 前期变换顶点缓存和后期变换顶点缓存1023.4 为提高性能交叉存放顶点数据1043.5 使用顶点数组或常量顶点数据1133.6 总结本章的最后一个示例1143.7 小结124第4章 小型JavaScript库与变换1274.1 JavaScript中矩阵和向量的操作1274.1.1 Sylvester库1284.1.2 WebGL-mjs库1324.1.3 glMatrix库1354.2 变换运算1394.3 理解完整的变换流水线1454.4 变换的实践1464.4.1 为对象坐标设置缓冲1474.4.2 用JavaScript创建变换矩阵并上传给着色器1484.4.3 将变换矩阵上传给GPU中的顶点着色器1484.4.4 调用绘图方法1494.5 理解变换顺序的重要性1504.5.1 使用一个固定的全局的坐标系1504.5.2 使用移动的局部的坐标系1534.5.3 变换矩阵的入栈和出栈操作1544.6 一个完整的示例：绘制几个变换后的对象1574.6.1 使用WebGL代码创建立方体1594.6.2 视图变换和模型变换的组织1614.7 小结162第5章 纹理贴图1635.1 理解丢失上下文1645.1.1 理解解决丢失上下文问题所需要的设置1645.1.2 处理丢失上下文问题时需要考虑的几个因素1665.2 2D纹理与立方映射纹理1695.3 载入纹理1705.3.1 创建WebGLTexture对象1705.3.2 绑定纹理1715.3.3 载入图像数据1715.3.4 将纹理上传到GPU1735.3.5 定义纹理参数1745.3.6 理解载入纹理的完整过程1755.3.7 创建一个纹理对象并载入纹理数据1775.4 定义纹理坐标1785.5 着色器中的纹理处理1805.6 处理纹理过滤1835.6.1 纹理伸展1845.6.2 纹理收缩1855.6.3 Mip映射纹理1865.7 理解纹理坐标包装1885.7.1 应用gl.REPEAT包装模式1885.7.2 应用gl.MIRRORED_REPEAT包装模式1905.7.3 应用gl.CLAMP_TO_EDGE包装模式1915.8 一个完整的应用纹理示例1915.9 获得用作纹理的图像1945.9.1 下载免费纹理1945.9.2 用自己拍摄的照片生成纹理1945.9.3 绘制图像1955.9.4 购买纹理1955.10 同域策略与跨域资源共享1955.10.1 同域策略应用于一般的图像1965.10.2 同域策略应用于纹理1975.10.3 跨域资源共享1995.11 小结200第6章 动画与用户输入2036.1 创建动画场景2036.1.1 setInterval()和setTimeout()的使用2056.1.2 使用requestAnimationFrame()函数2066.1.3 帧频不同引起的运动补偿2096.1.4 创建FPS计数器测量动画的平稳性2106.1.5 用FPS作为测量值的缺点2126.2 用户交互事件的处理2136.2.1 DOM Level0基本事件处理2146.2.2 DOM Level2——高级事件处理方法2156.2.3 键盘输入2176.2.4 鼠标输入2216.3 综合应用新知识2236.4 小结229第7章 光照2317.1 光源2317.2 局部光照模型的工作原理2327.3 Phong反射模型2327.3.1 环境反射2337.3.2 漫反射2347.3.3 镜面反射2367.3.4 Phong反射模型的完整公式和着色器2397.3.5 光照效果与纹理相结合2437.4 WebGL光照中需要的JavaScript代码2467.4.1 为顶点法线设置缓存2477.4.2 算法法线矩阵并上传给着色器2497.4.3 将光照信息上传给着色器2507.5 将不同的插值方法用于着色2507.5.1 平面着色2517.5.2 Gouraud着色2527.5.3 Phong着色2537.6 矢量必须归一化2567.6.1 顶点着色器中的矢量归一化2577.6.2 片段着色器的矢量归一化2577.7 应用不同类型的光源2587.7.1 平行光2587.7.2 点光源2597.7.3 聚光源2597.8 光强衰减2627.9 光照映射2657.10 小结267第8章 WebGL性能优化2698.1 WebGL底层工作机制2698.1.1 支持WebGL的硬件2708.1.2 关键的软件组成2718.2 WebGL性能优化2748.2.1 避免初学者的典型错误2748.2.2 确定瓶颈位置2758.2.3 有关性能的一般性建议2798.2.4 改善CPU受限的WebGL应用程序性能的建议2828.2.5 改善顶点受限

的WebGL应用程序性能的建议2838.2.6 改善像素受限的WebGL应用程序性能的建议2858.3 深入分析融合
2868.3.1 融合简介2868.3.2 设置融合函数2878.3.3 绘制顺序与深度缓冲区2908.3.4 绘制包含不透明对象
和半透明对象的场景2908.3.5 修改融合公式中的默认运算符2918.3.6 使用预乘alpha值2928.4 深入讨
论WebGL2928.4.1 使用WebGL框架2938.4.2 发布到Google Chrome WebStore2938.4.3 使用额外资源2938.5
小结294

编辑推荐

阿尤鲁等编著的《WebGL高级编程--开发Web3D图形》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，不需要阅读线性代码一节的全部内容(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。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>