

The WebGL Companion

book proposal for Taylor & Francis Group
philiprideout@gmail.com

Summary

Unique in the market by providing a focus on interactivity, *The WebGL Companion* guides readers through an essential set of rendering techniques and 3D interaction techniques, showcasing a set of small-but-complete web apps (*recipes*) at the end of each chapter, with diagrams and screenshots printed in full color. This book is not only the authoritative book on WebGL, it also serves as documentation for the *giza* library, developed over the course of the book.

About the Author

Philip Rideout has worked in the field of real-time graphics for over ten years, having played roles at several pioneering graphics companies, including Intergraph, NVIDIA, and Pixar. He is the sole author of *iPhone 3D Programming* (O'Reilly Media), and a contributing author of *GPU Pro 2* (A K Peters) and *OpenGL Insights* (CRC Press).

Tentative Outline

In the following outline, note that some section headings are colored in **green**; these are tutorial-style samples that demonstrate the techniques discussed in the chapter.

Contents

1	Preliminaries	6
1.1	A Brief History of *GL	6
	Describes WebGL's motivation, ancestry (OpenGL, OpenGL ES), and rapid growth. Also briefly mentions impediments at the time of writing (security concerns and IE support).	
1.2	Building Giza: Literate Programming	6
	Explains our coding conventions and the <code>giza</code> library that is developed over the course of the book.	
1.3	The Assembly Line Metaphor	6
	High-level overview of the WebGL rendering pipeline.	
1.4	The Canvas Element	6
	Explains the <code>width</code> and <code>height</code> attributes, how to handle retina displays.	
1.5	Animation Timing	6
	How to periodically trigger a draw events.	
1.6	Recipe 1: Strobe Light	6
	The simplest possible WebGL application; animates a solid color with <code>clear</code> .	
2	Vertex Shading and Transforms	7
2.1	Vector Algebra with Javascript	7
	Readers are assumed to have knowledge of elementary transforms. This section walks through <code>giza</code> 's implementation of vector and matrix classes.	
2.2	Life of a Vertex	7
	Describes model, view, and projection transforms.	
2.3	Shading Language Basics	7
	Writing <code>main</code> for vertex shaders.	
2.4	Line Drawing	7
	Explains the <code>LINES</code> primitive and <code>DrawElements</code> .	
2.5	Typed Arrays, Vertex Attributes and VBO's	7
	Shows how heterogeneous data (eg, colors and positions) can be interleaved and submitted to WebGL.	
2.6	Recipe 2: Color Graph	7
	Animated wireframe graph of the sinc function; illustrates performance differences between Javascript-side computation and shader-side computation	
3	Downloading and Applying Artistic Content	8
3.1	Texture Coordinates	8
	Texture coordinates and wrap modes; as an example, shows how to apply a section of the Mona Lisa to a quad.	

3.2	Texture Filtering	8
	Overview of minification, magnification, and mipmapping; uses small pixel art to illustrate the effects of filtering.	
3.3	Parametric Surfaces	8
	When the artist is the computer: procedural generation of simple geometry.	
3.4	Loading Mesh Data with XMLHttpRequest	8
	Implements Giza's download functionality; also mentions <code>webgl-loader</code> .	
3.5	Recipe 3: Planet Mars	8
	Downloads a sphere mesh and a Mars texture; also demonstrates <code>WEBGL_compressed_texture_s3tc</code> .	
4	Rendering and Lighting Solid Objects	9
4.1	Face Orientation	9
	Explains polygon winding, <code>GL_CULL_FACE</code> (used in the Planet Mars recipe) and <code>gl_FrontFacing</code>	
4.2	Dealing with Depth	9
	Explains the depth buffer and how depth artifacts can arise.	
4.3	Lambertian Reflection	9
	Illustrates a simple ambient-diffuse-specular lighting model.	
4.4	Recipe 4: Cel Shading	9
5	Framebuffer Effects, Part I	10
5.1	Alpha Blending	10
	Explains the different blend modes; also discusses back-to-front sorting and screen-door transparency with <code>discard</code> .	
5.2	Framebuffer Objects	10
	How to create offscreen render targets.	
5.3	Floating Point Textures	10
	Explains unclamped colors (<code>OES_texture_float</code>) and how they can be used for general-purpose computation.	
5.4	Subsurface Scattering	10
	Rendering translucent marble with floating point textures.	
5.5	Gaussian Blur	10
	Optimal filtering with minimal taps.	
5.6	Stencil Buffer	10
	Brief overview of stencil and how it can be used for CSG and shadows – almost mentions <code>WEBGL_depth_texture</code>	
5.7	Recipe 5: Mughal Window	10
	Illustrates stenciled fake reflection, a depth-based glass effect and a jaali screen backdrop with HDR Bloom	
6	Framebuffer Effects, Part II	11
6.1	Distance Blur	11
	Depth of field via post-processing; also mentions a jitter-based technique.	
6.2	Ambient Occlusion	11
	Screen-space ambient occlusion.	
6.3	Shadow Maps	11
	Old-school PCF shadow mapping.	

6.4	Heat Shimmer	11
	Using the fragment shader to create a “heat wave” effect.	
6.5	Recipe 6: Desert Scene	11
	Rotates the camera around a static desert scene that uses all the techniques discussed in this chapter.	
7	Texture Tricks, Part I	12
7.1	Point Sprites	12
	Show how to use point sprites and set their size dynamically.	
7.2	Derivatives Extension	12
	Shows how fragment shaders can smooth the edges of procedural patterns using <code>OES_standard_derivatives</code> .	
7.3	Recipe 7: Tube Sparks	12
	Fly through a tube with a stripe pattern, with embers falling onto the floor; animate fireworks in the vertex shader, rendered with additive blending	
8	Texture Tricks, Part II	13
8.1	Reflection and Refraction with Cubemaps	13
	Classic, simple use of cubemaps.	
8.2	Normal Mapping	13
	Bumps and dimples in tangent space.	
8.3	Using the WebCam	13
	Capturing video in real time and using it as a WebGL texture.	
8.4	Anisotropic Filtering	13
	High-quality rendering of nearly edge-on geometry using <code>EXT_texture_filter_anisotropic</code> .	
8.5	Recipe 8: Movie Theatre	13
	Displays a dimpled glass ball that refracts the webcam image.	
9	Interaction Techniques, Part I	14
9.1	Trackball Rotation	14
	Classic mouse-driven rotation.	
9.2	Selection Buffer	14
	Render object IDs into a FBO.	
9.3	Inverse Kinematics	14
	Shows how to animate an articulated skeleton. This is a high-level overview only, since giza’s IK implementation is quite math-heavy.	
9.4	Vertex Skinning	14
	Render soft joints by blending multiple vertex transformations.	
9.5	Recipe 9: Dancing Manikin	14
10	Interaction Techniques, Part II	15
10.1	View Cube	15
	Move the camera by clicking hotspots on a cube.	
10.2	Manipulators	15
	Interactive handles for positioning objects.	
10.3	Recipe 10: Set Dresser	15
	Place objects and spotlights into a scene.	

11 Beyond Giza: Related Libraries	16
11.1 three.js	16
By far the most popular 3D JavaScript library. It's higher level than giza, making it easy to create simple demos. It also provides support for non-WebGL rendering (eg, canvas or SVG rendering).	
11.2 glMatrix	16
Fast vector math.	
11.3 danser.js	16
Create music-based animation.	
11.4 tween.js	16
Robert Penner's animation equations.	

1

Preliminaries

- 1.1 A Brief History of *GL
- 1.2 Building Giza: Literate Programming
- 1.3 The Assembly Line Metaphor
- 1.4 The Canvas Element
- 1.5 Animation Timing
- 1.6 **Recipe 1: Strobe Light**

2

Vertex Shading and Transforms

- 2.1 Vector Algebra with Javascript
- 2.2 Life of a Vertex
- 2.3 Shading Language Basics
- 2.4 Line Drawing
- 2.5 Typed Arrays, Vertex Attributes and VBO's
- 2.6 **Recipe 2: Color Graph**

3

Downloading and Applying Artistic Content

- 3.1 Texture Coordinates
- 3.2 Texture Filtering
- 3.3 Parametric Surfaces
- 3.4 Loading Mesh Data with XMLHttpRequest
- 3.5 **Recipe 3: Planet Mars**

4

Rendering and Lighting Solid Objects

4.1 Face Orientation

4.2 Dealing with Depth

4.3 Lambertian Reflection

4.4 **Recipe 4: Cel Shading**

Applying standard lighting, then “snapping” the color gradient to a few colors for cartoon rendering.

5

Framebuffer Effects, Part I

- 5.1 Alpha Blending
- 5.2 Framebuffer Objects
- 5.3 Floating Point Textures
- 5.4 Subsurface Scattering
- 5.5 Gaussian Blur
- 5.6 Stencil Buffer
- 5.7 **Recipe 5: Mughal Window**

6

Framebuffer Effects, Part II

- 6.1 Distance Blur
- 6.2 Ambient Occlusion
- 6.3 Shadow Maps
- 6.4 Heat Shimmer
- 6.5 **Recipe 6: Desert Scene**

7

Texture Tricks, Part I

7.1 Point Sprites

7.2 Derivatives Extension

7.3 Recipe 7: Tube Sparks

8

Texture Tricks, Part II

- 8.1 Reflection and Refraction with Cube-maps
- 8.2 Normal Mapping
- 8.3 Using the WebCam
- 8.4 Anisotropic Filtering
- 8.5 **Recipe 8: Movie Theatre**

9

Interaction Techniques, Part I

9.1 Trackball Rotation

9.2 Selection Buffer

9.3 Inverse Kinematics

9.4 Vertex Skinning

9.5 **Recipe 9: Dancing Manikin**

Interact with a humanoid figure by dragging the parts.

10

Interaction Techniques, Part II

10.1 View Cube

10.2 Manipulators

10.3 Recipe 10: Set Dresser

11

Beyond Giza: Related Libraries

11.1 **three.js**

11.2 **glMatrix**

11.3 **danser.js**

11.4 **tween.js**