

# Contents

Preface

xvii

<b>1. Overview of WebGL</b>	<b>1</b>
Advantages of WebGL .....	3
You Can Start Developing 3D Graphics Applications Using Only a Text Editor.....	3
Publishing Your 3D Graphics Applications Is Easy.....	4
You Can Leverage the Full Functionality of the Browser .....	5
Learning and Using WebGL Is Easy.....	5
Origins of WebGL.....	5
Structure of WebGL Applications.....	6
Summary .....	7
<b>2. Your First Step with WebGL</b>	<b>9</b>
What Is a Canvas? .....	9
Using the <canvas> Tag .....	11
DrawRectangle.js .....	13
The World's Shortest WebGL Program: Clear Drawing Area .....	16
The HTML File (HelloCanvas.html).....	17
JavaScript Program (HelloCanvas.js).....	18
Experimenting with the Sample Program .....	23
Draw a Point (Version 1) .....	23
HelloPoint1.html.....	25
HelloPoint1.js.....	25
What Is a Shader?.....	27
The Structure of a WebGL Program that Uses Shaders .....	28
Initializing Shaders.....	30
Vertex Shader .....	33
Fragment Shader .....	35
The Draw Operation .....	36
The WebGL Coordinate System.....	38
Experimenting with the Sample Program .....	40
Draw a Point (Version 2) .....	41
Using Attribute Variables.....	41
Sample Program (HelloPoint2.js).....	42
Getting the Storage Location of an Attribute Variable .....	44
Assigning a Value to an Attribute Variable .....	45
Family Methods of gl.vertexAttrib3f() .....	47
Experimenting with the Sample Program .....	49

Draw a Point with a Mouse Click.....	50
Sample Program (ClickedPoints.js).....	50
Register Event Handlers .....	52
Handling Mouse Click Events.....	53
Experimenting with the Sample Program .....	57
Change the Point Color.....	58
Sample Program (ColoredPoints.js) .....	59
Uniform Variables.....	61
Retrieving the Storage Location of a Uniform Variable .....	62
Assigning a Value to a Uniform Variable .....	63
Family Methods of <code>gl.uniform4f()</code> .....	65
Summary .....	66

### **3. Drawing and Transforming Triangles 67**

Drawing Multiple Points.....	68
Sample Program (MultiPoint.js).....	70
Using Buffer Objects .....	72
Create a Buffer Object ( <code>gl.createBuffer()</code> ).....	74
Bind a Buffer Object to a Target ( <code>gl.bindBuffer()</code> ) .....	75
Write Data into a Buffer Object ( <code>gl.bufferData()</code> ).....	76
Typed Arrays.....	78
Assign the Buffer Object to an Attribute Variable ( <code>gl.vertexAttribPointer()</code> ).....	79
Enable the Assignment to an Attribute Variable ( <code>gl.enableVertexAttribArray()</code> ) ...	81
The Second and Third Parameters of <code>gl.drawArrays()</code> .....	82
Experimenting with the Sample Program .....	84
Hello Triangle .....	85
Sample Program (HelloTriangle.js) .....	85
Basic Shapes.....	87
Experimenting with the Sample Program .....	89
Hello Rectangle (HelloQuad) .....	89
Experimenting with the Sample Program .....	91
Moving, Rotating, and Scaling .....	91
Translation .....	92
Sample Program (TranslatedTriangle.js) .....	93
Rotation.....	96
Sample Program (RotatedTriangle.js).....	99
Transformation Matrix: Rotation .....	102
Transformation Matrix: Translation .....	105
Rotation Matrix, Again .....	106
Sample Program (RotatedTriangle_Matrix.js) .....	107

Reusing the Same Approach for Translation .....	111
Transformation Matrix: Scaling .....	111
Summary .....	113
<b>4. More Transformations and Basic Animation</b> .....	<b>115</b>
Translate and Then Rotate .....	115
Transformation Matrix Library: cuon-matrix.js .....	116
Sample Program (RotatedTriangle_Matrix4.js) .....	117
Combining Multiple Transformation .....	119
Sample Program (RotatedTranslatedTriangle.js) .....	121
Experimenting with the Sample Program .....	123
Animation .....	124
The Basics of Animation .....	125
Sample Program (RotatingTriangle.js) .....	126
Repeatedly Call the Drawing Function (tick()) .....	129
Draw a Triangle with the Specified Rotation Angle (draw()) .....	130
Request to Be Called Again (requestAnimationFrame()) .....	131
Update the Rotation Angle (animate()) .....	133
Experimenting with the Sample Program .....	135
Summary .....	136
<b>5. Using Colors and Texture Images</b> .....	<b>137</b>
Passing Other Types of Information to Vertex Shaders .....	137
Sample Program (MultiAttributeSize.js) .....	139
Create Multiple Buffer Objects .....	140
The gl.vertexAttribPointer() Stride and Offset Parameters .....	141
Sample Program (MultiAttributeSize_Interleaved.js) .....	142
Modifying the Color (Varying Variable) .....	146
Sample Program (MultiAttributeColor.js) .....	147
Experimenting with the Sample Program .....	150
Color Triangle (ColoredTriangle.js) .....	151
Geometric Shape Assembly and Rasterization .....	151
Fragment Shader Invocations .....	155
Experimenting with the Sample Program .....	156
Functionality of Varying Variables and the Interpolation Process .....	157
Pasting an Image onto a Rectangle .....	160
Texture Coordinates .....	162
Pasting Texture Images onto the Geometric Shape .....	162
Sample Program (TexturedQuad.js) .....	163
Using Texture Coordinates (initVertexBuffers()) .....	166
Setting Up and Loading Images (initTextures()) .....	166
Make the Texture Ready to Use in the WebGL System (loadTexture()) .....	170

Flip an Image's Y-Axis.....	170
Making a Texture Unit Active (gl.activeTexture()).....	171
Binding a Texture Object to a Target (gl.bindTexture()).....	173
Set the Texture Parameters of a Texture Object (gl.texParameteri()).....	174
Assigning a Texture Image to a Texture Object (gl.texImage2D()).....	177
Pass the Texture Unit to the Fragment Shader (gl.uniform1i()) .....	179
Passing Texture Coordinates from the Vertex Shader to the Fragment Shader ...	180
Retrieve the Texel Color in a Fragment Shader (texture2D()) .....	181
Experimenting with the Sample Program .....	182
Pasting Multiple Textures to a Shape.....	183
Sample Program (MultiTexture.js) .....	184
Summary .....	189
<b>6. The OpenGL ES Shading Language (GLSL ES)</b> .....	<b>191</b>
Recap of Basic Shader Programs .....	191
Overview of GLSL ES .....	192
Hello Shader!.....	193
Basics .....	193
Order of Execution.....	193
Comments.....	193
Data (Numerical and Boolean Values) .....	194
Variables .....	194
GLSL ES Is a Type Sensitive Language.....	195
Basic Types.....	195
Assignment and Type Conversion.....	196
Operations.....	197
Vector Types and Matrix Types.....	198
Assignments and Constructors .....	199
Access to Components.....	201
Operations.....	204
Structures .....	207
Assignments and Constructors .....	207
Access to Members .....	207
Operations.....	208
Arrays .....	208
Samplers .....	209
Precedence of Operators .....	210
Conditional Control Flow and Iteration.....	211
if Statement and if-else Statement.....	211
for Statement.....	211
continue, break, discard Statements.....	212

Functions .....	213
Prototype Declarations.....	214
Parameter Qualifiers.....	214
Built-In Functions.....	215
Global Variables and Local Variables.....	216
Storage Qualifiers.....	217
const Variables.....	217
Attribute Variables.....	218
Uniform Variables.....	218
Varying Variables.....	219
Precision Qualifiers.....	219
Preprocessor Directives.....	221
Summary .....	223

**7. Toward the 3D World** **225**

What's Good for Triangles Is Good for Cubes.....	225
Specifying the Viewing Direction.....	226
Eye Point, Look-At Point, and Up Direction.....	227
Sample Program (LookAtTriangles.js).....	229
Comparing LookAtTriangles.js with RotatedTriangle_Matrix4.js.....	232
Looking at Rotated Triangles from a Specified Position.....	234
Sample Program (LookAtRotatedTriangles.js).....	235
Experimenting with the Sample Program.....	236
Changing the Eye Point Using the Keyboard.....	238
Sample Program (LookAtTrianglesWithKeys.js).....	238
Missing Parts.....	241
Specifying the Visible Range (Box Type).....	241
Specify the Viewing Volume.....	242
Defining a Box-Shaped Viewing Volume.....	243
Sample Program (OrthoView.html).....	245
Sample Program (OrthoView.js).....	246
Modifying an HTML Element Using JavaScript.....	247
The Processing Flow of the Vertex Shader.....	248
Changing Near or Far.....	250
Restoring the Clipped Parts of the Triangles (LookAtTrianglesWithKeys_ViewVolume.js).....	251
Experimenting with the Sample Program.....	253
Specifying the Visible Range Using a Quadrangular Pyramid.....	254
Setting the Quadrangular Pyramid Viewing Volume.....	256
Sample Program (PerspectiveView.js).....	258
The Role of the Projection Matrix.....	260
Using All the Matrices (Model Matrix, View Matrix, and Projection Matrix).....	262

Sample Program (PerspectiveView_mvp.js) .....	263
Experimenting with the Sample Program .....	266
Correctly Handling Foreground and Background Objects .....	267
Hidden Surface Removal .....	270
Sample Program (DepthBuffer.js) .....	272
Z Fighting .....	273
Hello Cube .....	275
Drawing the Object with Indices and Vertices Coordinates.....	277
Sample Program (HelloCube.js) .....	278
Writing Vertex Coordinates, Colors, and Indices to the Buffer Object.....	281
Adding Color to Each Face of a Cube.....	284
Sample Program (ColoredCube.js) .....	285
Experimenting with the Sample Program .....	287
Summary .....	289
<b>8. Lighting Objects</b> .....	<b>291</b>
Lighting 3D Objects.....	291
Types of Light Source.....	293
Types of Reflected Light.....	294
Shading Due to Directional Light and Its Diffuse Reflection .....	296
Calculating Diffuse Reflection Using the Light Direction and the Orientation of a Surface.....	297
The Orientation of a Surface: What Is the Normal? .....	299
Sample Program (LightedCube.js) .....	302
Add Shading Due to Ambient Light .....	307
Sample Program (LightedCube_ambient.js) .....	308
Lighting the Translated-Rotated Object.....	310
The Magic Matrix: Inverse Transpose Matrix.....	311
Sample Program (LightedTranslatedRotatedCube.js) .....	312
Using a Point Light Object .....	314
Sample Program (PointLightedCube.js).....	315
More Realistic Shading: Calculating the Color per Fragment.....	319
Sample Program (PointLightedCube_perFragment.js) .....	319
Summary .....	321
<b>9. Hierarchical Objects</b> .....	<b>323</b>
Drawing and Manipulating Objects Composed of Other Objects .....	324
Hierarchical Structure.....	325
Single Joint Model.....	326
Sample Program (JointModel.js) .....	328
Draw the Hierarchical Structure (draw()) .....	332
A Multijoint Model .....	334

Sample Program (MultiJointModel.js) .....	335
Draw Segments (drawBox()).....	339
Draw Segments (drawSegment()).....	340
Shader and Program Objects: The Role of initShaders().....	344
Create Shader Objects (gl.createShader()).....	345
Store the Shader Source Code in the Shader Objects (g.shaderSource()).....	346
Compile Shader Objects (gl.compileShader()).....	347
Create a Program Object (gl.createProgram()).....	349
Attach the Shader Objects to the Program Object (gl.attachShader()).....	350
Link the Program Object (gl.linkProgram()).....	351
Tell the WebGL System Which Program Object to Use (gl.useProgram()).....	353
The Program Flow of initShaders() .....	353
Summary .....	356

**10. Advanced Techniques 357**

Rotate an Object with the Mouse.....	357
How to Implement Object Rotation.....	358
Sample Program (RotateObject.js) .....	358
Select an Object .....	360
How to Implement Object Selection .....	361
Sample Program (PickObject.js).....	362
Select the Face of the Object.....	365
Sample Program (PickFace.js).....	366
HUD (Head Up Display) .....	368
How to Implement a HUD.....	369
Sample Program (HUD.html).....	369
Sample Program (HUD.js) .....	370
Display a 3D Object on a Web Page (3DoverWeb) .....	372
Fog (Atmospheric Effect) .....	372
How to Implement Fog.....	373
Sample Program (Fog.js).....	374
Use the w Value (Fog_w.js) .....	376
Make a Rounded Point .....	377
How to Implement a Rounded Point .....	377
Sample Program (RoundedPoints.js).....	378
Alpha Blending .....	380
How to Implement Alpha Blending .....	380
Sample Program (LookAtBlendedTriangles.js).....	381
Blending Function.....	382
Alpha Blend 3D Objects (BlendedCube.js) .....	384
How to Draw When Alpha Values Coexist .....	385

Switching Shaders .....	386
How to Implement Switching Shaders .....	387
Sample Program (ProgramObject.js) .....	387
Use What You've Drawn as a Texture Image .....	392
Framebuffer Object and Renderbuffer Object .....	392
How to Implement Using a Drawn Object as a Texture .....	394
Sample Program (FramebufferObject.js) .....	395
Create Frame Buffer Object (gl.createFramebuffer()) .....	397
Create Texture Object and Set Its Size and Parameters .....	397
Create Renderbuffer Object (gl.createRenderbuffer()) .....	398
Bind Renderbuffer Object to Target and Set Size (gl.bindRenderbuffer(), gl.renderbufferStorage()) .....	399
Set Texture Object to Framebuffer Object (gl.bindFramebuffer(), gl.framebufferTexture2D()) .....	400
Set Renderbuffer Object to Framebuffer Object (gl.framebufferRenderbuffer()) .....	401
Check Configuration of Framebuffer Object (gl.checkFramebufferStatus()) .....	402
Draw Using the Framebuffer Object .....	403
Display Shadows .....	405
How to Implement Shadows .....	405
Sample Program (Shadow.js) .....	406
Increasing Precision .....	412
Sample Program (Shadow_highp.js) .....	413
Load and Display 3D Models .....	414
The OBJ File Format .....	417
The MTL File Format .....	418
Sample Program (OBJViewer.js) .....	419
User-Defined Object .....	422
Sample Program (Parser Code in OBJViewer.js) .....	423
Handling Lost Context .....	430
How to Implement Handling Lost Context .....	431
Sample Program (RotatingTriangle_contextLost.js) .....	432
Summary .....	434
<b>A. No Need to Swap Buffers in WebGL</b> .....	<b>437</b>
<b>B. Built-in Functions of GLSL ES 1.0</b> .....	<b>441</b>
Angle and Trigonometry Functions .....	441
Exponential Functions .....	443
Common Functions .....	444
Geometric Functions .....	447



---

Matrix Functions.....	448
Vector Functions.....	449
Texture Lookup Functions.....	451
<b>C. Projection Matrices</b>	<b>453</b>
Orthogonal Projection Matrix.....	453
Perspective Projection Matrix.....	453
<b>D. WebGL/OpenGL: Left or Right Handed?</b>	<b>455</b>
Sample Program <i>CoordinateSystem.js</i> .....	456
Hidden Surface Removal and the Clip Coordinate System.....	459
The Clip Coordinate System and the Viewing Volume.....	460
What Is Correct?.....	462
Summary .....	464
<b>E. The Inverse Transpose Matrix</b>	<b>465</b>
<b>F. Load Shader Programs from Files</b>	<b>471</b>
<b>G. World Coordinate System Versus Local Coordinate System</b>	<b>473</b>
The Local Coordinate System.....	474
The World Coordinate System.....	475
Transformations and the Coordinate Systems.....	477
<b>H. Web Browser Settings for WebGL</b>	<b>479</b>
<b>Glossary</b>	<b>481</b>
<b>References</b>	<b>485</b>
<b>Index</b>	<b>487</b>