

Contents

Foreword	xxi	
Barthold Lichtenbelt		
Preface	xxiii	
Tips	xxvii	
I Discovering	1	
1 Teaching Computer Graphics Starting with Shader-Based OpenGL	3	OpenGL
Edward Angel		
1.1 Introduction	3	
1.2 A Basic Course	4	
1.3 Hello World in OpenGL: Old Style	4	
1.4 Starting with Programmable Pipelines	6	
1.5 Hello World: New Style	8	
1.6 The Rest of the Course	10	
1.7 Conclusion	15	
Bibliography	16	
2 Transitioning Students to Post-Deprecation OpenGL	17	OpenGL
Mike Bailey		
2.1 Introduction	17	
2.2 Naming Shader Variables: Introduction	18	
2.3 Naming Shader Variables: Details	18	
2.4 Indexed Vertex Buffer Object C++ Class	20	
2.5 GLSLProgram C++ Class	23	
2.6 Conclusion	26	
Bibliography	26	

WebGL for OpenGL Developers	27
Patrick Cozzi and Scott Hunter	
3.1 Introduction	27
3.2 The Benefits of WebGL	28
3.3 Security	34
3.4 Deploying Shaders	36
3.5 The JavaScript Language	37
3.6 Resources	45
Bibliography	45
Porting Mobile Apps to WebGL	47
Ashraf Samy Hegab	
4.1 Introduction	47
4.2 OpenGL across Platforms	47
4.3 Getting Started	48
4.4 Loading Textures	51
4.5 Camera and Matrices	53
4.6 Controls	54
4.7 Other Considerations	56
4.8 Maintenance	57
4.9 Conclusion	58
Bibliography	59
The GLSL Shader Interfaces	61
Christophe Riccio	
5.1 Introduction	61
5.2 Variables and Blocks	62
5.3 Locations	64
5.4 Matching Interfaces	68
5.5 Working with Semantics	77
5.6 Application-Side Validations for Debug Build Only	82
5.7 Conclusion	85
Bibliography	86
An Introduction to Tessellation Shaders	87
Philip Rideout and Dirk Van Gelder	
6.1 Introduction	87
6.2 The New Shading Pipeline	90
6.3 Tessellating a Teapot	99
6.4 Isolines and Spirals	102
6.5 Incorporating Other OpenGL Features	103
Bibliography	104

7 Procedural Textures in GLSL	105	OpenGL WebGL OpenGL ES
Stefan Gustavson		
7.1 Introduction	105	
7.2 Simple Functions	107	
7.3 Antialiasing	108	
7.4 Perlin Noise	110	
7.5 Worley Noise	113	
7.6 Animation	115	
7.7 Texture Images	115	
7.8 Performance	118	
7.9 Conclusion	119	
Bibliography	119	
8 OpenGL SC Emulation Based on OpenGL and OpenGL ES	121	OpenGL OpenGL ES OpenGL SC
Hwanyong Lee and Nakhoon Baek		
8.1 Introduction	121	
8.2 OpenGL SC Implementations	123	
8.3 Design and Implementation	125	
8.4 Results	129	
8.5 Conclusion	131	
Bibliography	131	
9 Mixing Graphics and Compute with Multiple GPUs	133	OpenGL
Alina Alt		
9.1 Introduction	133	
9.2 Graphics and Compute Interoperability on an API Level	134	
9.3 Graphics and Compute Interoperability on a System Level	138	
9.4 Conclusion	142	
Bibliography	142	
II Rendering Techniques	143	
10 GPU Tessellation: We Still Have a LOD of Terrain to Cover	145	OpenGL
António Ramires Fernandes and Bruno Oliveira		
10.1 Introduction	145	
10.2 Rendering Terrains with OpenGL GPU Tessellation	146	
10.3 A Simple Approach to Dynamic Level of Detail	149	
10.4 Roughness: When Detail Matters	152	
10.5 Crunching Numbers, or Is This All That Matters?	153	
10.6 Conclusion	160	
Bibliography	160	

nGL bGL LES	11 Antialiased Volumetric Lines Using Shader-Based Extrusion	163
	Sébastien Hillaire	
	11.1 Introduction	163
	11.2 Antialiased Lines Using Postprocess Antialiasing	164
	11.3 Antialiased Volumetric Lines Using Geometry Extrusion	165
	11.4 Performance	169
	11.5 Conclusion	170
	Bibliography	171
nGL bGL LES	12 2D Shape Rendering by Distance Fields	173
	Stefan Gustavson	
	12.1 Introduction	173
	12.2 Method Overview	174
	12.3 Better Distance Fields	176
	12.4 Distance Textures	176
	12.5 Hardware Accelerated Distance Transform	177
	12.6 Fragment Rendering	177
	12.7 Special Effects	179
	12.8 Performance	180
	12.9 Shortcomings	180
	12.10 Conclusion	181
	Bibliography	182
ebGL	13 Efficient Text Rendering in WebGL	183
	Benjamin Encz	
	13.1 Introduction	183
	13.2 Canvas-Based Font Rendering	184
	13.3 Bitmap Font Rendering	187
	13.4 Comparison	190
	13.5 Conclusion	194
	Bibliography	194
enGL	14 Layered Textures Rendering Pipeline	195
	Dzmitry Malyshau	
	14.1 Introduction	195
	14.2 Layered Pipeline	197
	14.3 Results	202
	Bibliography	204

	15 Depth of Field with Bokeh Rendering	205	OpenGL
	Charles de Rousiers and Matt Pettineo		
	15.1 Introduction	205	
	15.2 Depth of Field Phenomemon	206	
	15.3 Related Work	208	
	15.4 Algorithm	209	
	15.5 Results	214	
	15.6 Discussion	216	
	15.7 Conclusion	216	
	Bibliography	217	
	16 Shadow Proxies	219	OpenGL
	Jochem van der Spek		
	16.1 Introduction	219	
	16.2 Anatomy of a Shadow Proxy	221	
	16.3 Setting Up the Pipeline	223	
	16.4 The ShadowProxy-Enabled Fragment Shader	224	
	16.5 Modulating the Shadow Volume	226	
	16.6 Performance	227	
	16.7 Conclusion and Future Work	228	
	Bibliography	228	
	III Bending the Pipeline	229	
	17 Real-Time Physically Based Deformation Using Transform Feedback	231	OpenGL
	Muhammad Mobeen Movania and Lin Feng		
	17.1 Introduction	231	
	17.2 Hardware Support and Evolution of Transform Feedback	232	
	17.3 The Mechanism of Transform Feedback	233	
	17.4 Mathematical Model	234	
	17.5 Implementation	238	
	17.6 Experimental Results and Comparisons	244	
	17.7 Conclusion	245	
	Bibliography	245	
	18 Hierarchical Depth Culling and Bounding-Box Management on the GPU	247	OpenGL
	Dzmitry Malyshau		
	18.1 Introduction	247	
	18.2 Pipeline	248	

18.3 Order of Operations	254
18.4 Experimental Results	255
18.5 Conclusion and Future Work	257
Bibliography	257

19 Massive Number of Shadow-Casting Lights with Layered Rendering 259

Daniel Rákos

19.1 Introduction	259
19.2 Traditional Shadow Map Rendering in OpenGL	260
19.3 Our Shadow Map Generation Algorithm	263
19.4 Performance	265
19.5 Advanced Techniques	275
19.6 Limitations	276
19.7 Conclusion	277
Bibliography	278

20 Efficient Layered Fragment Buffer Techniques 279

Pyarelal Knowles, Geoff Leach, and Fabio Zambetta

20.1 Introduction	279
20.2 Related Work	281
20.3 The Linked-List LFB	282
20.4 The Linearized LFB	283
20.5 Performance Results	286
20.6 Conclusion	290
Bibliography	290

21 Programmable Vertex Pulling 293

Daniel Rákos

21.1 Introduction	293
21.2 Implementation	294
21.3 Performance	296
21.4 Application	298
21.5 Limitations	299
21.6 Conclusion	300
Bibliography	300

22 Octree-Based Sparse Voxelization Using the GPU Hardware Rasterizer 303

Cyril Crassin and Simon Green

22.1 Introduction	303
22.2 Previous Work	304

22.3 Unrestricted Memory Access in GLSL	305
22.4 Simple Voxelization Pipeline	306
22.5 Sparse Voxelization into an Octree	312
22.6 Conclusion	317
Bibliography	317

IV Performance 321

23 Performance Tuning for Tile-Based Architectures 323

Bruce Merry

23.1 Introduction	323
23.2 Background	324
23.3 Clearing and Discarding the Framebuffer	326
23.4 Incremental Frame Updates	328
23.5 Flushing	329
23.6 Latency	330
23.7 Hidden Surface Removal	332
23.8 Blending	333
23.9 Multisampling	333
23.10 Performance Profiling	334
23.11 Summary	335
Bibliography	335

24 Exploring Mobile vs. Desktop OpenGL Performance 337

Jon McCaffrey

24.1 Introduction	337
24.2 Important Differences and Constraints	338
24.3 Reducing Memory Bandwidth	341
24.4 Reducing Fragment Workload	345
24.5 Vertex Shading	349
24.6 Conclusion	350
Bibliography	351

25 Improving Performance by Reducing Calls to the Driver 353

Sébastien Hillaire

25.1 Introduction	353
25.2 Efficient OpenGL States Usage	354
25.3 Batching and Instancing	357
25.4 Conclusion	362
Bibliography	362

WebGL
OpenGL ES

OpenGL
WebGL
OpenGL ES

OpenGL

26 Indexing Multiple Vertex Arrays	365
Arnaud Masserann	
26.1 Introduction	365
26.2 The Problem	365
26.3 An Algorithm	368
26.4 Vertex Comparison Methods	369
26.5 Performance	371
26.6 Conclusion	373
Bibliography	373

27 Multi-GPU Rendering on NVIDIA Quadro	375
Shalini Venkataraman	
27.1 Introduction	375
27.2 Previous Scaling Approaches	376
27.3 Targeting a Specific GPU for Rendering	377
27.4 Optimized Data Transfer between GPUs	382
27.5 Application Structure for Multi-GPU	383
27.6 Parallel Rendering Methodologies	385
27.7 Conclusion	388
Bibliography	388

V Transfers **389**

28 Asynchronous Buffer Transfers	391
Ladislav Hrabcak and Arnaud Masserann	
28.1 Introduction	391
28.2 Buffer Objects	392
28.3 Upload	398
28.4 Download	402
28.5 Copy	405
28.6 Multithreading and Shared Contexts	405
28.7 Usage Scenario	408
28.8 Conclusion	413
Bibliography	414

29 Fermi Asynchronous Texture Transfers	415
Shalini Venkataraman	
29.1 Introduction	415
29.2 OpenGL Command Buffer Execution	417

29.3 Current Texture Transfer Approaches	418
29.4 GPU Asynchronous Texture Transfers	421
29.5 Implementation Details	423
29.6 Results and Analysis	425
29.7 Conclusion	429
Bibliography	430

30 WebGL Models: End-to-End **431** **WebGL**

Won Chun

30.1 Introduction	431
30.2 Life of a 3D Model	432
30.3 A Coherent Whole	441
30.4 Key Improvements	447
30.5 Conclusion	452
Bibliography	452

31 In-Game Video Capture with Real-Time Texture Compression **455** **OpenGL**

Brano Kemen

31.1 Introduction	455
31.2 Overview of DXT Compression	456
31.3 DXT Compression Algorithms	456
31.4 Transformation to YUV Style Color Spaces	458
31.5 Comparison	460
31.6 Using Real-Time DXT Compression for Procedural Content and Video Capture	462
31.7 Conclusion	465
Bibliography	466

32 An OpenGL-Friendly Geometry File Format and Its Maya Exporter **467** **OpenGL**

Adrien Herubel and Venceslas Biri

32.1 Introduction	467
32.2 Manifesto	468
32.3 The Drone Format	469
32.4 Writing a Maya File Translator	474
32.5 Results	477
32.6 Conclusion	479
Bibliography	479

Debugging and Profiling**481****ARB_debug_output: A Helping Hand for Desperate Developers****483**

Antônio Ramires Fernandes and Bruno Oliveira

33.1 Introduction	483
33.2 Exposing the Extension	484
33.3 Using a Callback Function	484
33.4 Sorting Through the Cause of Events	485
33.5 Accessing the Message Log	486
33.6 Adding Custom User Events to the Log	487
33.7 Controlling the Event Output Volume	488
33.8 Preventing Impact on the Final Release	488
33.9 Clash of the Titans: Implementation Strategies	489
33.10 Further Thoughts on Debugging	491
33.11 Conclusion	491
Bibliography	492

The OpenGL Timer Query**493**

Christopher Lux

34.1 Introduction	493
34.2 Measuring OpenGL Execution Times	495
34.3 Conclusion	501
Bibliography	502

A Real-Time Profiling Tool**503**

Lionel Fuentes

35.1 Introduction	503
35.2 Scope and Requirements	504
35.3 Design of the Tool	505
35.4 Implementation	507
35.5 Using the Profiler	510
35.6 Conclusions	512
Bibliography	512

Browser Graphics Analysis and Optimizations**513**

Chris Dirks and Omar A. Rodriguez

36.1 Introduction	513
36.2 The Stages of Bloom	514
36.3 Overhead of Bloom	515
36.4 Analyzing WebGL Applications	516
36.5 Analysis Workflow on Windows	519

36.6 Optimized Bloom	522
36.7 Conclusion	526
Bibliography	526

37 Performance State Tracking**527**

OpenGL

Aleksandar Dimitrijević

37.1 Introduction	527
37.2 Power Consumption Policies	528
37.3 P-State Tracking Using NVAPI	528
37.4 P-State Tracking Using ADL	532
37.5 Conclusion	533
Bibliography	534

38 Monitoring Graphics Memory Usage**535**

OpenGL

Aleksandar Dimitrijević

38.1 Introduction	535
38.2 Graphics Memory Allocation	536
38.3 Querying Memory Status on NVIDIA Cards	537
38.4 Querying Memory Status on AMD Cards	538
38.5 Conclusion	540
Bibliography	540

VII Software Design**541****39 The ANGLE Project: Implementing OpenGL ES 2.0 on Direct3D****543**WebGL
OpenGL ES

Daniel Koch and Nicolas Capens

39.1 Introduction	543
39.2 Background	543
39.3 Implementation	544
39.4 Future Work	568
39.5 Conclusion	568
39.6 Source Code	569
Bibliography	569

40 SceneJS: A WebGL-Based Scene Graph Engine**571**

WebGL

Lindsay Kay

40.1 Introduction	571
40.2 Efficiently Abstracting WebGL	572
40.3 Optimizing the Scene	578

40.4 Picking	580
40.5 Conclusion	581
Bibliography	581
41 Features and Design Choices in SpiderGL	583
Marco Di Benedetto, Fabio Ganovelli, and Francesco Banterle	
41.1 Introduction	583
41.2 Library Architecture	584
41.3 Representing 3D Objects	585
41.4 Direct Access to WebGL Object State	590
41.5 WebGLObject Wrappers	598
41.6 Conclusion	602
Bibliography	603
42 Multimodal Interactive Simulations on the Web	605
Tansel Halic, Woojin Ahn, Suvranu De	
42.1 Introduction	605
42.2 II-SoFMIS Design and Definitions of Modules	606
42.3 Framework Implementation	607
42.4 Rendering Module	609
42.5 Simulation Module	611
42.6 Hardware Module	613
42.7 Case Study: LAGB Simulator	614
42.8 Conclusion	619
Bibliography	619
43 A Subset Approach to Using OpenGL and OpenGL ES	621
Jesse Barker and Alexandros Frantzis	
43.1 Introduction	621
43.2 Making Legacy Code Modern	622
43.3 Keeping Code Maintainable across API Variants	626
43.4 What if I Need a Specific Piece of Functionality?	633
43.5 Conclusion	633
Bibliography	634
44 The Build Syndrome	635
Jochem van der Spek and Daniel Dekkers	
44.1 Introduction	635
44.2 Using Utility Libraries	637
44.3 OpenGL Agnosticism	640
44.4 Configuration Spaces	642
44.5 Metabuilds and CMake	643

44.6 CMake and the Configuration Space	644
44.7 CMake and Platform Specifics	648
44.8 Conclusion	654
Bibliography	654
About the Contributors	657
Index	673