Game Engine Gems 2

Edited by Eric Lengyel

Game Engine Gems 2

Edited by Eric Lengyel

A K Peters, Ltd.
Natick, Massachusetts

Editorial, Sales, and Customer Service Office

A K Peters, Ltd.
5 Commonwealth Rd, Suite 2C
Natick, MA 01760
www.akpeters.com

ISBN 978-1-56881-437-7

Library of Congress Control No. 2010003949

Cover image: *Splint/Second* screenshot © 2010 Disney.

Printed in India
15 14 13 12 11

10 9 8 7 6 5 4 3 2 1

Game Engine Gems 2

Preface

The word *gem* has been coined in the fields of computer graphics and game development as a term for describing a short article that focuses on a particular technique, a clever trick, or practical advice that a person working in these fields would find interesting and useful. Several book series containing the word "Gems" in their titles have appeared since the early 1990s, and we continued the tradition by establishing the *Game Engine Gems* series in 2010.

This book is the second volume of the *Game Engine Gems* series, and it comprises a collection of new game engine development techniques. A group of 29 experienced professionals, several of whom also contributed to the first volume, have written down portions of their knowledge and wisdom in the form of the 31 chapters that follow.

The topics covered in these pages vary widely within the subject of game engine development and have been divided into the three broad categories of graphics and rendering, game engine design, and systems programming. The first part of the book presents a variety of rendering techniques and dedicates four entire chapters to the increasingly popular topic of stereoscopic rendering. The second part contains several chapters that discuss topics relating to the design of large components of a game engine. The final part of the book presents several gems concerning topics of a "low-level" nature for those who like to work with the nitty-gritty of engine internals.

Audience

The intended audience for this book includes professional game developers, students of computer science programs, and practically anyone possessing an interest in how the pros tackle specific problems that arise during game engine

development. Many of the chapters assume a basic knowledge of computer architecture as well as some knowledge of the high-level design of current-generation game consoles, such as the PlayStation 3 and Xbox 360. The level of mathematics used in the book rarely exceeds that of basic trigonometry and calculus.

The Website

The official website for the *Game Engine Gems* series can be found at the following address:

http://www.gameenginegems.net/

Supplementary materials for many of the gems in this book are posted on this website, and they include demos, source code, examples, specifications, and larger versions of some figures. For chapters that include project files, the source code can be compiled using Microsoft Visual Studio.

Any corrections to the text that may arise will be posted on the website. This is also the location at which proposals will be accepted for the next volume in the *Game Engine Gems* series.

Acknowledgements

Many thanks are due to A K Peters for quickly assuming ownership of the *Game Engine Gems* series after it had lost its home well into the period during which contributing authors had been writing their chapters. Of course, thanks also go to these contributors, who took the transition in stride and produced a great set of gems for this volume. They all worked hard during the editing process so that we would be able to stay on the original schedule.

Contents

Preface xv

Part I Graphics and Rendering 1

Chapter 1 Fast Computation of Tight-Fitting Oriented Bounding Boxes 3
Thomas Larsson Linus Källberg
1.1 Introduction 3
1.2 Algorithm 4
1.3 Evaluation 10
1.4 Optimization Using SIMD Instructions 16
1.5 Discussion and Future Work 17

Chapter 2 Modeling, Lighting, and Rendering Techniques for Volumetric Clouds 21
Frank Kane
2.1 Modeling Cloud Formation 22
2.2 Cloud Lighting Techniques 27
2.3 Cloud Rendering Techniques 30

Chapter 3 Simulation of Night-Vision and Infrared Sensors 45
Frank Kane
3.1 The Physics of the Infrared 45
3.2 Simulating Infrared Sensor Effects 49
3.3 Night-Vision Goggle Simulation 51

Chapter 4 Screen-Space Classification for Efficient Deferred Shading **55**

Balor Knight Matthew Ritchie George Parrish

4.1 Introduction 55
4.2 Overview of Method 56
4.3 Depth-Related Classification 58
4.4 Pixel Classification 60
4.5 Combining Classification Results 61
4.6 Index Buffer Generation 62
4.7 Tile Rendering 64
4.8 Shader Management 64
4.9 Platform Specifics 66
4.10 Optimizations 69
4.11 Performance Comparison 72

Chapter 5 Delaying OpenGL Calls **75**

Patrick Cozzi

5.1 Introduction 75
5.2 Motivation 75
5.3 Possible Implementations 77
5.4 Delayed Calls Implementation 78
5.5 Implementation Notes 83
5.6 Improved Flexibility 83
5.7 Concluding Remarks 84

Chapter 6 A Framework for GLSL Engine Uniforms **87**

Patrick Cozzi

6.1 Introduction 87
6.2 Motivation 87
6.3 Implementation 89
6.4 Beyond GLSL Built-in Uniforms 95
6.5 Implementation Tips 95
6.6 Concluding Remarks 96

Chapter 7 A Spatial and Temporal Coherence Framework for Real-Time Graphics **97**
Michal Drobot
7.1 Introduction 98
7.2 The Spatiotemporal Framework 100
7.3 Applications 109
7.4 Future Work 115

Chapter 8 Implementing a Fast DDOF Solver **119**
Holger Grün
8.1 Introduction 119
8.2 Modifying the Basic CR Solver 123
8.3 Results 131

Chapter 9 Automatic Dynamic Stereoscopic 3D **135**
Jason Hughes
9.1 General Problems in S3D 135
9.2 Problems in S3D Unique to Games 138
9.3 Dynamic Controls 140
9.4 A Simple Dynamic S3D Camera 141
9.5 Content-Adaptive Feedback 143

Chapter 10 Practical Stereo Rendering **151**
Matthew Johnson
10.1 Introduction to Stereo 3D 151
10.2 Overview of Stereo Displays 152
10.3 Introduction to Rendering Stereo 153
10.4 The Mathematics of Stereo Views and Projection 154
10.5 Using Geometry Shader to Render Stereo Pairs 159

Chapter 11 Making 3D Stereoscopic Games **163**
Sébastien Schertenleib
11.1 Introduction 163
11.2 How Stereoscopic 3D Works 163
11.3 How to Set Up the Virtual 3D Cameras 164
11.4 Safe Area 166
11.5 Technical Considerations 169
11.6 Same Scene, Both Eyes, and How to Optimize 169
11.7 Scene Traversal 172

11.8 Supporting Both Monoscopic and Stereoscopic Versions 173
11.9 Visual Quality 173
11.10 Going One Step Further 177
11.11 Conclusion 177

Chapter 12 A Generic Multiview Rendering Engine Architecture 179
M. Adil Yalçın Tolga Çapın
12.1 Introduction 179
12.2 Analyzing Multiview Displays 180
12.3 The Architecture 182
12.4 The Multiview Camera 185
12.5 The Multiview Buffer 189
12.6 The Multiview Compositor 190
12.7 Rendering Management 191
12.8 Rendering Optimizations 193
12.9 Discussion 195

Chapter 13 3D in a Web Browser 199
Rémi Arnaud
13.1 A Brief History 199
13.2 Fast Forward 204
13.3 3D with Flash 207
13.4 3D with Java 212
13.5 3D with a Game Engine Plug-In 215
13.6 Google Native Client 218
13.7 3D with HTML5 220
13.8 Conclusion 224

Chapter 14 2D Magic 229
Daniel Higgins
14.1 Tools of the Trade 229
14.2 Position 230
14.3 Color and Opacity 232
14.4 Texture (UV) Coordinates 234
14.5 What a Mesh! 236
14.6 Mesh Architecture 237
14.7 Mesh Examples 239
14.8 Conclusion 248

Part II Game Engine Design **249**

Chapter 15 High-Performance Programming with Data-Oriented Design **251**
Noel Llopis
15.1 Modern Hardware 251
15.2 Principles of Data-Oriented Design 253
15.3 Data-Oriented Design Benefits 254
15.4 How to Apply Data-Oriented Design 255
15.5 Real-World Situations 256
15.6 Parallelization 259
15.7 Conclusion 261

Chapter 16 Game Tuning Infrastructure **263**
Wessam Bahnassi
16.1 Introduction 263
16.2 The Need for Tweak 263
16.3 Design Considerations 264
16.4 The Tuning Tool 265
16.5 Data Exchange 269
16.6 Schema and Exposure 270
16.7 Data Storage 271
16.8 Case Studies 273
16.9 Final Words 276

Chapter 17 Placeholders beyond Static Art Replacement **279**
Olivier Vaillancourt Richard Egli
17.1 Placeholder Assets in a Game 279
17.2 Preaching by Example: The Articulated Placeholder Model 285
17.3 Integration in a Production Environment 301
17.4 In the End, Is It Really Needed? 303
17.5 Implementation 304

Chapter 18 Believable Dead Reckoning for Networked Games **307**
Curtiss Murphy
18.1 Introduction 307
18.2 Fundamentals 307
18.3 Pick an Algorithm, Any Algorithm 310
18.4 Time for *T* 313

18.5 Publish or Perish 315
18.6 Ground Clamping 320
18.7 Orientation 322
18.8 Advanced Topics 324
18.9 Conclusion 326

Chapter 19 An Egocentric Motion Management System 329
Michael Ramsey
19.1 Fundamental Components of the ECMMS 331
19.2 Collision Sensors 331
19.3 Query Space 332
19.4 Modeling the Environment 334
19.5 The ECMMS Architecture 335
19.6 Modeling an ECMMS-Enabled Agent 335
19.7 Generating a Behavior Model with the ECMMS 336
19.8 Animation Validation 339
19.9 A Single Agent Behavioral Response Algorithm and Example 339

Chapter 20 Pointer Patching Assets 343
Jason Hughes
20.1 Introduction 343
20.2 Overview of the Technique 346
20.3 A Brief Example 348

Chapter 21 Data-Driven Sound Pack Loading and Organization 355
Simon Franco
21.1 Introduction 355
21.2 Constructing a Sound Map 356
21.3 Constructing Sound Packs by Analyzing the Event Table 358
21.4 Constructing and Using Sound Loading Triggers 361
21.5 Conclusion 363

Chapter 22 GPGPU Cloth Simulation Using GLSL, OpenCL, and CUDA 365
Marco Fratarcangeli
22.1 Introduction 365
22.2 Numerical Algorithm 366
22.3 Collision Handling 368
22.4 CPU Implementation 369
22.5 GPU Implementations 371

22.6 GLSL Implementation 372
22.7 CUDA Implementation 373
22.8 OpenCL Implementation 374
22.9 Results 375
22.10 Future Work 377
22.11 Demo 378

Chapter 23 A Jitter-Tolerant Rigid Body Sleep Condition 379
Eric Lengyel
23.1 Introduction 379
23.2 The Sleep Condition 380

Part III Systems Programming 383

Chapter 24 Bit Hacks for Games 385
Eric Lengyel
24.1 Integer Sign Manipulation 385
24.2 Predicates 388
24.3 Miscellaneous Tricks 392
24.4 Logic Formulas 395

Chapter 25 Introspection for C++ Game Engines 397
Jon Watte
25.1 Introduction 397
25.2 The Demo 400
25.3 The Gem 401
25.4 Lifting the Veil 404
25.5 To and From a Network 405
25.6 In Closing 407

Chapter 26 A Highly Optimized Portable Memory Manager 409
Jason Hughes
26.1 Introduction 409
26.2 Overview 411
26.3 Small Block Allocator 415
26.4 Medium Block Allocator 418
26.5 Large Block Allocator 422

26.6 Page Management in the Memory Manager 422
26.7 OSAPI Ideas 425

Chapter 27 Simple Remote Heaps 427
Jason Hughes
27.1 Introduction 427
27.2 Bitwise Remote Heap 428
27.3 Blockwise Remote Heap 430
27.4 Testing Results 435

Chapter 28 A Cache-Aware Hybrid Sorter 437
Manny Ko
28.1 Stream Splitting 438
28.2 Substream Sorting 440
28.3 Stream Merging and Loser Tree 442
28.4 Multicore Implementation 445
28.5 Conclusion 446
Appendix 447

Chapter 29 Thread Communication Techniques 449
Julien Hamaide
29.1 Latency and Threading 449
29.2 Single Writer, Single Reader 450
29.3 The Aggregator 453
29.4 The Dispatcher 455
29.5 The Gateway 455
29.6 Debugging 456

Chapter 30 A Cross-Platform Multithreading Framework 457
Martin Fleisz
30.1 Threading 457
30.2 Synchronization Objects 459
30.3 Limitations 471
30.4 Future Extensions 473

Chapter 31 Producer-Consumer Queues 475
Matthew Johnson
31.1 Introduction 475
31.2 Multithreading Overview 477

31.3 A First Approach: Using Win32 Semaphores and Critical Sections 477
31.4 A Second Approach: Lock-Free Algorithms 482
31.5 Processor Architecture Overview and Memory Models 483
31.6 Lock-Free Algorithm Design 486
31.7 Lock-Free Implementation of a Free List 487
31.8 Lock-Free Implementation of a Queue 491
31.9 Interprocess Communication 496

Contributor Biographies **499**

Index **509**

Part I

Graphics and Rendering