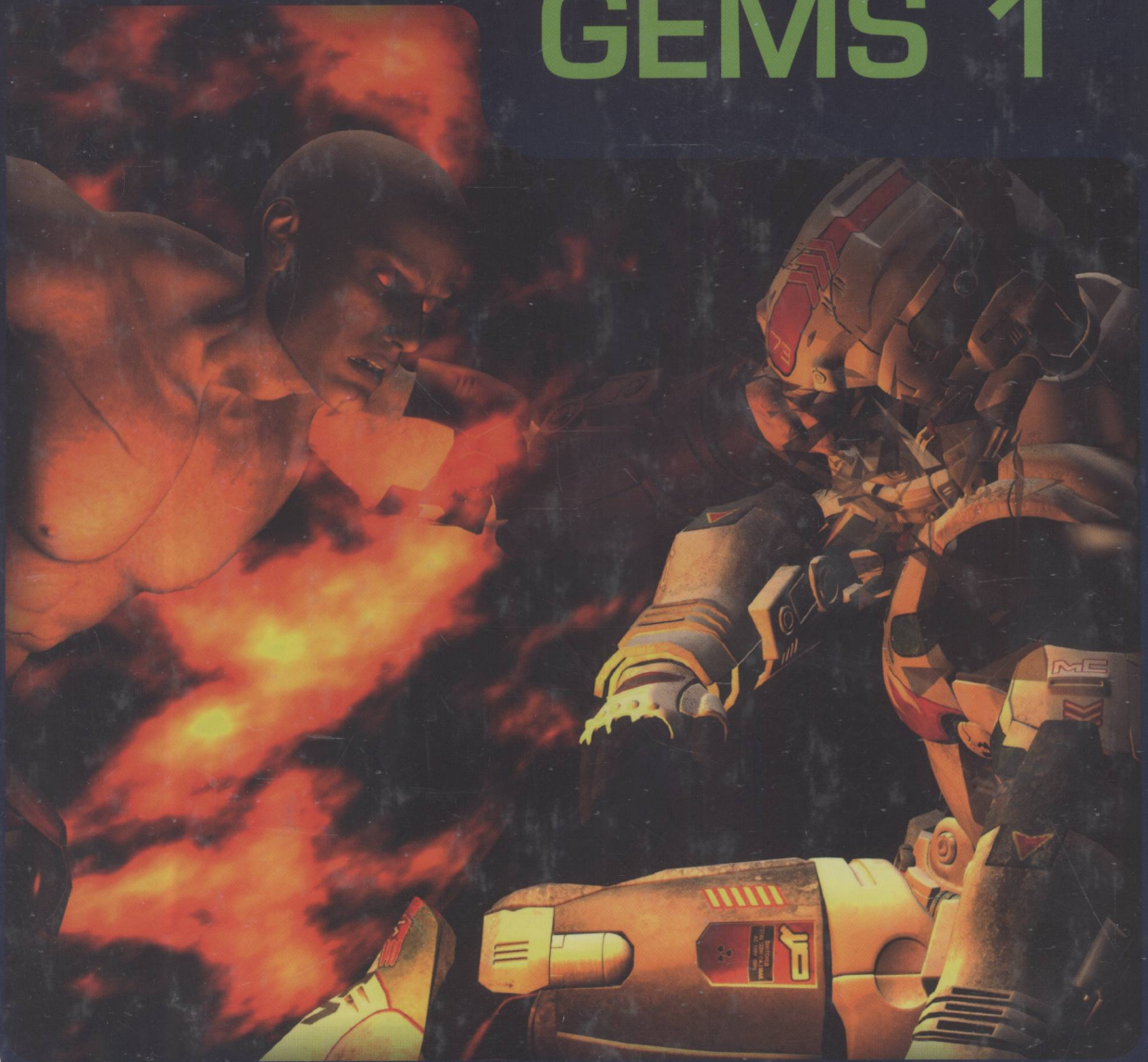


# GAME ENGINE

## GEMS 1



SERIES EDITOR: ERIC LENGYEL

TS952.83  
G192  
v.1

# Game Engine Gems

## Volume One

Edited by

Eric Lengyel, Ph.D.



E2010002466



**JONES AND BARTLETT PUBLISHERS**

*Sudbury, Massachusetts*

BOSTON

TORONTO

LONDON

SINGAPORE

附光盘壹张

*World Headquarters*  
Jones and Bartlett Publishers  
40 Tall Pine Drive  
Sudbury, MA 01776  
978-443-5000  
info@jbpub.com  
www.jbpub.com

Jones and Bartlett Publishers Canada  
6339 Ormindale Way  
Mississauga, Ontario L5V 1J2  
Canada

Jones and Bartlett Publishers International  
Barb House, Barb Mews  
London W6 7PA  
United Kingdom

Jones and Bartlett's books and products are available through most bookstores and online booksellers. To contact Jones and Bartlett Publishers directly, call 800-832-0034, fax 978-443-8000, or visit our website, [www.jbpub.com](http://www.jbpub.com).

Substantial discounts on bulk quantities of Jones and Bartlett's publications are available to corporations, professional associations, and other qualified organizations. For details and specific discount information, contact the special sales department at Jones and Bartlett via the above contact information or send an email to [specialsales@jbpub.com](mailto:specialsales@jbpub.com).

Copyright © 2011 by Jones and Bartlett Publishers, LLC

ISBN-13: 9780763778880  
ISBN-10: 0763778885

All rights reserved. No part of the material protected by this copyright may be reproduced or utilized in any form, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without written permission from the copyright owner.

The publisher recognizes and respects all marks used by companies, manufacturers, and developers as a means to distinguish their products. All brand names and product names mentioned in this book are trademarks or service marks of their respective companies. Any omission or misuse (of any kind) of service marks or trademarks, etc., is not an attempt to infringe on the copyright of others.

### **Production Credits**

Publisher: David Pallai  
Editorial Assistant: Molly Whitman  
Senior Production Editor: Katherine Crighton  
Associate Marketing Manager: Lindsay Ruggiero  
V.P., Manufacturing and Inventory Control: Therese Connell  
Cover Design: Kristin E. Parker  
Cover Image: © Jesse-lee Lang/Dreamstime.com  
Printing and Binding: Malloy, Inc.  
Cover Printing: Malloy, Inc.

6048

Printed in the United States of America

14 13 12 11 10 10 9 8 7 6 5 4 3 2 1

---

## Introduction

In the fields of computer graphics and computer game development, the word *gem* has been established 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. The term *gem* was first used in 1990 for the first volume of the *Graphics Gems* series of books, which concentrated on knowledge pertaining to computer graphics. The mainstream methods for rendering 3D images have changed considerably since then, but many of those gems still comprise useful techniques today and have demonstrated a timeless quality to the knowledge they contain. Several newer book series containing the word “Gems” in their titles have appeared in related subject areas such as game programming and GPU rendering, and they all advance the notion of sharing knowledge through concise articles that each focus on a specific topic. We continue the tradition with this book, the first volume of *Game Engine Gems*.

*Game Engine Gems* concentrates on knowledge relating to the development of game engines, which encompass the architecture, design, and coding methods constituting the technological foundation for today’s video games. A complete game engine typically includes large components that handle graphics, audio, networking, and physics. There may also be large components that provide services for artificial intelligence (AI) and graphical user interfaces (GUIs), as well as a variety of smaller components that deal with resource management, input devices, mathematics, multithreading, and many additional pieces of generic functionality required by the games built upon them. Furthermore, many game engines are able to run on multiple platforms, which may include PCs and one or more game consoles such as the PlayStation 3 or Xbox 360. The *Game Engine Gems* series is specifically intended to include all such aspects of game engine development targeting all current game platforms.

This book is divided into three parts covering the broad subject areas of game engine design, rendering techniques, and programming methods. The 28 gems appearing in this book are written by a group of 25 authors having expertise in game engine development, some quite extensive. It is our hope that the wisdom recorded in these pages and the pages of future volumes of *Game Engine Gems* continue to serve game developers for many years to come.

## Call for Papers

---

At the time this book is published, work on the second volume of *Game Engine Gems* will have already entered its early stages. If you are a professional developer working in a field related to game development and would like to submit a contribution to the next book in the series, please visit our official website at <http://www.gameenginegems.com/>.

---

## Contributor Biographies

---

### Rémi Arnaud

[remi@acm.org](mailto:remi@acm.org)

Rémi Arnaud is working as Chief Software Architect at Screampoint International, a company providing interoperable 5D digital city models for the benefit of governments, property owners, developers, designers, contractors, managers, and service providers. Rémi's involvement with real-time graphics started in the R&D department of Thomson Training & Simulation (now Thales) designing and then leading the Space Magic real-time visual system for training simulators, where he finalized his Ph.D. "La synthèse d'images en temps réel". He then relocated to California to join the Silicon Graphics IRIS Performer team, working on advanced features such as calligraphic light points for training pilots. He then decided to be more adventurous and co-founded Intrinsic Graphics, where he co-designed the Alchemy engine, a middleware targeting cross-platform game development for PS2, Xbox, GameCube, and PC. He was hired as Graphics Architect at Sony Computer Entertainment US R&D, working on the PlayStation 3 SDK graphics API, and joined the Khronos Group to create COLLADA asset exchange standard. More recently, Rémi worked at Intel where he created and lead the Larrabee Game Engine Technology team.

---

### Ron Barbosa

[ron@exibeo.net](mailto:ron@exibeo.net)

Ron Barbosa has been an avid hobbyist game and game technology developer since his teenage years. Since 1993, he has worked as a professional network/software engineer for many companies producing internet technologies, including former technology giants Compaq Computer Corporation and Lucent Technologies, Inc. He currently serves as the Chief Software Architect at Boca Raton, Florida's Revellex Corporation, a travel technology services provider. In his short spurts of spare time, he attempts to remain active in indie game development circles and is the original author of *Planet Crashmania 9,000,000* available on Microsoft's Xbox

LIVE Indie Games service and Apple's iPod Touch Apps Store (ported to iPod Touch by James Webb).

---

**John Bolton****johnjbolton@yahoo.com**

---

John Bolton is a software engineer at Netflix in Los Gatos, California and has been programming games professionally since 1992. He has contributed to dozens of games and has been lead programmer on several titles, including *I Have No Mouth and I Must Scream*, *Heroes of Might and Magic*, and *High Heat Baseball*.

---

**Khalid Djado****Khalid.Djado@USherbrooke.ca**

---

Khalid Djado is a Ph.D. student in the Department of Computer Sciences at University of Sherbrooke. His research interests include computer graphics and physical simulations. He is a lecturer for graduate students in game development for the University of Sherbrooke at Ubisoft Campus. He was also a game developer at Amusement Cyanide in Montreal. He obtained a bachelor's degree in applied mathematics from the University Sidi Mohamed Ben Abdellah in Morocco, and a master's in modelling, simulation, and optimisation from the University of Bretagne Sud in France. He has been a member of ACM Siggraph since 2006.

---

**Richard Egli****Richard.Egli@USherbrooke.ca**

---

Richard Egli is professor in the Department of Computer Sciences at University of Sherbrooke since 2000. He received his B.Sc. degree in Computer Science and his M.Sc. degree in Computer Sciences at University of Sherbrooke (Québec, Canada). He received his Ph.D. in Computer Sciences from University of Montréal (Québec, Canada) in 2000. He is the director of the centre MOIVRE (MODélisation en Imagerie, Vision et RÉseaux de neurones). His research interests include computer graphics, physical simulations, and digital image processing.

---

**Simon Franco****simon\_franco@hotmail.com**

---

Simon Franco sampled his first taste of programming on the Commodore Amiga, when he wrote his first *Pong* clone in AMOS, and he has been coding ever since. He joined the games industry in 2000 after completing a degree in computer science. He started at The Creative Assembly in 2004, where he has been to this day. When he's not playing the latest game, he'll be writing assembly code for the ZX spectrum.

**Anders Hast****aht@cb.uu.se**

Anders Hast works half time as a Visualization Expert at UPPMAX (Uppsala Multidisciplinary Center for Advanced Computational Science) and half time as associate professor at the University of Gävle, both in Sweden. He has published well over 50 scientific papers in journals, in conferences, and as book chapters in various areas in computer graphics, visualization, and applied mathematics. His other interests in life, besides computer graphics research, are US model trains, drinking Czech beer, and studying the Italian language.

**Daniel F. Higgins****webmaster@programming.org**

Dan has spent over 10 years in the games industry, starting with Stainless Steel Studios. He was one of the original creators of the Titan game engine, and was one of the chief AI programmers on *Empire Earth*, *Empires: Dawn of the Modern World* and *Rise & Fall: Civilizations at War*. Later, he worked at Tilted Mill on *Caesar IV* and *SimCity Societies*. Today, along with his wife, he is owner and manager of Lunchtime Studios, Inc.

**Adrian Hirst****adrian@weaseltron.com**

Adrian Hirst has been shedding blood, sweat, and tears programming on any and every gaming platform for the last ten years, working with many leading developers and publishers, most recently including Sony, Codemasters (*LMA 2002*, *Colin McRae PC 3, 4, 5*, 2005+), and Electronic Arts/Criterion (*Burnout: Paradise*). Most recently he set up Weaseltron Entertainment in order to join the growing masses of independent developers and apply his skills to new challenges. He is also remarkably good looking, writes his own biography, and needs a beer.

**Jason Hughes****jhughes@steelpennygames.com**

Jason Hughes is an industry veteran game programmer of 15 years and has been actively coding for 25 years. His background covers everything from modem drivers in 6502 assembly to fluid dynamics on the Wii to a multi-platform 3D engine. Jason tinkers with exotic data structures, advanced compression algorithms, and various tools and technology relating to the games industry. Prior to founding Steel Penny Games, Jason spent several years at Naughty Dog on the ICE team writing the asset pipeline tools used by PS3 developers in the ICE and Edge libraries.

**Frank Kane****fkane@sundog-soft.com**

Frank Kane is the owner of Sundog Software, LLC, makers of the *SilverLining* SDK for real-time rendering of skies, clouds, and precipitation effects (see [www.sundog-soft.com](http://www.sundog-soft.com) for more information). Frank's game development experience began at Sierra On-Line, where he worked on the system-level software of a dozen classic adventure game titles including *Phantasmagoria*, *Gabriel Knight II*, *Police Quest: SWAT*, and *Quest for Glory V*. He's also an alumnus of Looking Glass Studios, where he helped develop *Flight Unlimited III*. Frank developed the C2Engine scene rendering engine for SDS International's Advanced Technology Division, which is used for virtual reality training simulators by every branch of the US military. He currently lives with his family outside Seattle.

**Jan Krassnigg****Jan@Krassnigg.de**

Jan Krassnigg is studying Information Technologies at the University of Aachen, Germany.

**Martin Linklater****mslinklater@mac.com**

Martin Linklater has been programming since 1981 when he was ten years old. After spending his teenage years hacking C64 and Amiga code, he got a Bachelors Degree in Computer Science in 1993. His first job in the games industry was as a programmer for Psygnosis, soon to become Sony Computer Entertainment Europe. After five years at SCEE he left with five colleagues to start Curly Monsters, an independent development house. Curly Monsters closed in 2003 after releasing two titles. Martin worked for a short time for EA, then returned to Sony in 2003. Martin is currently a Technical Director working on an undisclosed Sony title. Martin lives in Wallasey, UK with his wife and two-year-old son. He enjoys games, flight simulation, sim racing, and beer.

**Colt McAnlis****duhroach@gmail.com**

Colt McAnlis is a graphics programmer at Blizzard Entertainment, where he works on stuff he typically can't talk about. Prior, Colt was a graphics programmer at Microsoft Ensemble studios, where in his free time he moonlighted as an Adjunct Professor at SMU's GUILDHALL school for video game development. He has received a number of publications in various industry books, and continues to be an active participant in speaking at conferences.

**Jeremy Moore****jeremy.moore@disney.com**

Jeremy Moore is the lead engine programmer for the Core Technology Group at Disney's Black Rock Studio in Brighton, UK. He has been working in the games industry for over a decade. Four of those years were spent working on SCEA's *ATV Offroad Fury* games on both PS2 and PSP. Among other things, he was responsible for the acclaimed network play implementation. He now specializes in real-time graphics and being ordered around by his two young daughters.

**Jon Parise****jon@indelible.org**

Jon Parise is a senior software engineer at Electronic Arts. He has worked on a number of titles, including *The Sims 3*, *The Lord of the Rings: The White Council*, *Ultima Online*, and *The Sims Online*. He was also a contributing author for *Massively Multiplayer Game Development 2*. Jon earned a bachelors degree in Information Technology from the Rochester Institute of Technology and a masters degree in Entertainment Technology from Carnegie Mellon University.

**Kurt Pelzer****kurt.pelzer@gmx.net**

Kurt Pelzer is a Senior Software Engineer and Software Architect with a decade of experience in team-oriented projects within the 3D real-time simulation and games industry. At Piranha Bytes, he has taken part in the development of the games *Risen* (PC & Xbox 360), *Gothic 1-3* (PC) and the engine technology used for these products. Kurt has published articles in the technical book series *GPU Gems*, *Game Programming Gems*, and *ShaderX*.

**Aurelio Reis****AurelioReis@gmail.com**

Aurelio Reis is a programmer at id Software, where he works on graphics and special effects. While he's interested in all aspects of game development, he especially enjoys working on networking and gameplay as well as doing research on cutting edge graphics techniques. An industry veteran and avid gamer, Aurelio has contributed to numerous titles over the years, but is most excited about the game he's working on right now, *Doom 4*.

**Sébastien Schertenleib****sscherten@bluewin.ch**

Sébastien Schertenleib has been involved in academic research projects creating 3D mixed reality systems using stereoscopic visualization while completing his Ph.D. in Computer Graphics at the Swiss Institute of Technology in Lausanne. Since then, he has been holding a job as a Principal Engineer at Sony Computer Enter-

tainment Europe's R&D Division. This role includes supporting game developers on all PlayStation platforms by providing technical training, presenting at various games conferences, and working directly with game developers via on-site technical visits and code share.

---

**László Szirmay-Kalos****szirmay@iit.bme.hu**

László Szirmay-Kalos is the head of the Department of Control Engineering and Information Technology at the Budapest University of Technology and Economics. He received his Ph.D. in 1992 and full professorship in 2001 in computer graphics. His research area is Monte-Carlo global illumination algorithms and their GPU implementation. He has more than two hundred publications in this field. He is the fellow of *Eurographics*.

---

**Balázs Tóth****tbalazs@sch.bme.hu**

Balázs Tóth is an assistant processor at the Budapest University of Technology and Economics. He is involved in distributed GPGPU projects and deferred shading rendering and is responsible for the CUDA education of the faculty.

---

**Tamás Umenhoffer****umitomi@gmail.com**

Tamás Umenhoffer is an assistant processor at the Budapest University of Technology and Economics. His research topic is the computation of global illumination effects and realistic lighting in participation media and their application in real-time systems and games.

---

**Brad Werth****bradley.j.werth@intel.com**

Brad Werth is a Senior Software Engineer in Intel's Visual Computing Division. He has been a frequent speaker at the Game Developers Conference and Austin GDC.

---

**David Williams****david@david-williams.info**

David Williams received his M.Sc. in Computer Science from the University of Warwick in 2004 before joining City University as a Ph.D. student researching Medical Visualization. It was at this point that he developed an interest in voxels and began investigating how the concepts could be applied to game engines. He received his Ph.D. in 2008, but has continued to work on his Thermite3D voxel engine in his spare time. He now works as a graphics programmer for a game development company in the UK, and also enjoys photography and travelling.

---

## About the Editor

**Eric Lengyel**

**[lengyel@terathon.com](mailto:lengyel@terathon.com)**

---

Eric Lengyel is a veteran of the computer games industry with over 15 years of experience writing game engines. He has a Ph.D. in Computer Science from the University of California, Davis, and he has a Masters Degree in Mathematics from Virginia Tech. Eric is the founder of Terathon Software, where he currently leads ongoing development of the C4 Engine.

Eric entered the games industry at the Yosemite Entertainment division of Sierra Online in Oakhurst, California, where he was the lead programmer for the fifth installment of the popular adventure RPG series *Quest for Glory*. He then worked on the OpenGL team for Apple Computer at their headquarters in Cupertino, California. More recently, Eric worked in the Advanced Technology Group at Naughty Dog in Santa Monica, California, where he designed graphics driver software used on the PlayStation 3 game console.

Eric is the author of the bestselling book *Mathematics for 3D Game Programming and Computer Graphics*. He is also the author of *The OpenGL Extensions Guide*, the mathematical concepts chapter in the book *Introduction to Game Development*, and several articles in the *Game Programming Gems* series. His articles have also been published in the *Journal of Game Development*, in the *Journal of Graphics Tools*, and on *Gamasutra.com*. Eric currently serves on the editorial board for the recently renamed *Journal of Graphics, GPU, and Game Tools* (JGGGT).

---

# Contents

<b>Introduction</b>	<b>xiii</b>
<b>Contributor Biographies</b>	<b>xv</b>
<b>About the Editor</b>	<b>xxi</b>

---

## **Part I Game Engine Design** **1**

---

### **Chapter 1 What to Look for When Evaluating Middleware for Integration** **3**

*Jason Hughes*

1.1 Middleware, How Do I Love Thee?	3
1.2 Integration Complexity and Modularity	3
1.3 Memory Management	4
1.4 Mass Storage I/O Access	5
1.5 Logging	5
1.6 Error Handling	6
1.7 Stability and Performance Consistency	6
1.8 Custom Profiling Tools	7
1.9 Customer Support	7
1.10 Demands on the Maintainers	7
1.11 Source Code Availability	8
1.12 Quality of Source Code	8

1.13 Platform Portability	8
1.14 Licensing Requirements	9
1.15 Cost	10
<b>Chapter 2 The Game Asset Pipeline</b>	<b>11</b>
<i>Rémi Arnaud</i>	
2.1 Asset Pipeline Overview	11
2.2 Asset Pipeline Design	18
2.3 Push or Pull Pipeline Model	20
2.4 COLLADA, A Standard Intermediate Language	22
2.5 OpenCOLLADA	30
2.6 User Content	33
<b>Chapter 3 Volumetric Representation of Virtual Environments</b>	<b>39</b>
<i>David Williams</i>	
3.1 Introduction	39
3.2 Overview	40
3.3 Data Structures	42
3.4 Surface Extraction	46
3.5 Rendering	52
3.6 Physics	56
3.7 The Future	57
<b>Chapter 4 High-Level Pathfinding</b>	<b>61</b>
<i>Daniel Higgins</i>	
4.1 Terms	62
4.2 Start Your Engines	62
4.3 Why High-Level Pathfinding?	63
4.4 Preprocess Phase	64
4.5 Fuzzy Pathing Phase	71
4.6 Detailed Paths Phase	75
4.7 Why Go Through All This Trouble?	76

<b>Chapter 5    Environment Sound Culling</b>	<b>79</b>
<i>Simon Franco</i>	
5.1 The Problem	79
5.2 A Sound Culling Solution	80
5.3 Constructing the Sound Grid	83
5.4 Processing the Sound Grid	84
5.5 Supporting Multiple Listeners	90
5.6 Extensions	90
 <b>Chapter 6    A GUI Framework and Presentation Layer</b>	 <b>91</b>
<i>Adrian Hirst</i>	
6.1 GUI Systems	91
6.2 Design Patterns: Model View Controller (MVC)	92
6.3 A GUI Design	93
6.4 And Finally	104
 <b>Chapter 7    World's Best Palettizer</b>	 <b>107</b>
<i>Jason Hughes</i>	
7.1 Palettes? Whatever for?	107
7.2 Understanding Quantization	108
7.3 Hard-Earned Lessons	109
7.4 Algorithm Overview	111
7.5 Future Work	120
7.6 Results	120
 <b>Chapter 8    3D Stereoscopic Rendering: An Overview of Implementation Issues</b>	 <b>123</b>
<i>Anders Hast</i>	
8.1 Mechanisms of Plano-Stereoscopic Viewing	124
8.2 Stereo Techniques	131
8.3 Design Considerations for 3D Scenes	133
8.4 Outlook	135

<b>Chapter 9    A Multithreaded 3D Renderer</b>	<b>139</b>
<i>Sebastien Schertenleib</i>	
9.1 The Memory Model	140
9.2 Building the Display Lists in Parallel	141
9.3 Parallel Models	143
9.4 Synchronizing the GPU and CPU	144
9.5 Using Additional Processing Resources	145
9.6 Reducing the Pressure on the Memory Bandwidth	145
9.7 Performing Graphical Operations in Parallel	146
<b>Chapter 10   Camera-Centric Engine Design for Multithreaded Rendering</b>	<b>149</b>
<i>Colt McAnlis</i>	
10.1 Uses of Multi-Core in Video Games	150
10.2 Multithreaded Command Buffers	151
10.3 Device-Independent Command Buffers	152
10.4 A Camera-Centric Design	157
10.5 Future Work	164
<b>Chapter 11   A GPU-Managed Memory Pool</b>	<b>167</b>
<i>Jeremy Moore</i>	
11.1 Background	168
11.2 The Memory Pool	169
11.3 Synchronization Issues	170
11.4 The Staging Buffer	172
11.5 Memory Pool Defragmentation	173
11.6 Memory Pool Eviction	174
11.7 Platform-Specific Considerations	174
11.8 Future Work	175
<b>Chapter 12   Precomputed 3D Velocity Field for Simulating Fluid Dynamics</b>	<b>177</b>
<i>Khalid Djado and Richard Egli</i>	
12.1 Introduction	177

12.2 Velocity Field Computation	178
12.3 Physics Simplification	181
12.4 Results and Discussion	184
<b>Chapter 13 Mesh Partitioning for Fun and Profit</b>	<b>187</b>
<i>Jason Hughes</i>	
13.1 Desirable Algorithm Properties	187
13.2 Lessons Learned	189
13.3 When Greedy Is Good	191
13.4 Future Work	193
13.5 Graphical Walkthrough	194
<b>Chapter 14 Moments of Inertia for Common Shapes</b>	<b>197</b>
<i>Eric Lengyel</i>	
14.1 Center of Mass	197
14.2 The Inertia Tensor	198
14.3 Derivation of Moments of Inertia	201
14.4 Summary	215
<b>Part II Rendering Techniques</b>	<b>217</b>
<hr/>	
<b>Chapter 15 Physically-Based Outdoor Scene Lighting</b>	<b>219</b>
<i>Frank Kane</i>	
15.1 Positioning the Sun and Moon	219
15.2 Computing Natural Sunlight	221
15.3 Moonlight and Other Nighttime Light Sources	223
15.4 Tone-Mapping the Light	224
15.5 Implementation Notes	226
<b>Chapter 16 Rendering Physically-Based Skyboxes</b>	<b>229</b>
<i>Frank Kane</i>	