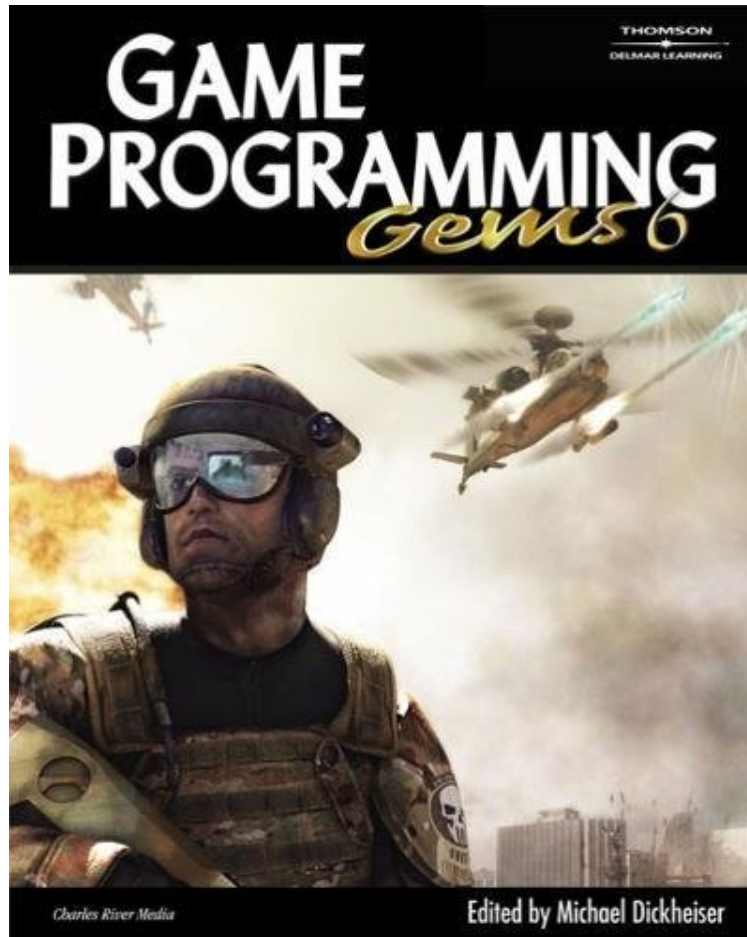


[Read now] Game Programming Gems 6 (Book CD-ROM) (Game Development Series)

## Game Programming Gems 6 (Book CD-ROM) (Game Development Series)

*Mike Dickheiser*

*ePub | \*DOC | audiobook | ebooks | Download PDF*



[Download](#)

[Read Online](#)

#606256 in Books Dickheiser, Michael (EDT) 2006-03-07Ingredients: Example IngredientsOriginal language:EnglishPDF # 1 9.50 x 7.75 x 1.50l, 3.28 #File Name: 1584504501700 pages | File size: 52.Mb

**Mike Dickheiser : Game Programming Gems 6 (Book CD-ROM) (Game Development Series)** before purchasing it in order to gage whether or not it would be worth my time, and all praised Game Programming Gems 6 (Book CD-ROM) (Game Development Series):

53 of 54 people found the following review helpful. Good bag of tricks for advanced game programmersBy calvinmeThis is the latest volume in the game programming gems series that is now six years old. This book, like the others in the series, is a collection of articles written by industry experts for the advanced game programmer. Because it covers so much ground with little or no background information, I doubt that one reader will be able to appreciate the whole book. The book comes with a CD that has source code for all of the articles, which is very helpful.I'm a multimedia programmer rather than a game programmer, and I found quite a few nuggets in this edition. Among them was the article on using OpenCV, which is the open source computer vision library, and also the article on CPPUnit,

which can be used for unit testing of any type of software, not just games. The section on math and physics was the least helpful. It either set the bar too low with easy articles like "How to do floating point faster than hardware", or set the bar too high with difficult terse articles such as "Real-Time Particle-Based Fluid Simulation". However, if you are into game physics, this section might be something you really enjoy. In the AI section, it seemed like I had seen most of the material before. The last article though was really unique - "Constructing a Goal-Oriented Robot for UnrealTournament Using Fuzzy Sensors, Finite-state Behaviors and Behavior Networks". This is an outstanding article for those interested in robots in games. The whole section on scripting was worthwhile. I especially enjoyed the Script language survey at the front of the section, since this is a moving target with many languages to choose from. The graphics section is quite advanced, and unless "Realtime Rendering" seems like light bedtime reading to you, most of the articles will probably not make much sense. My favorite section of the book was the one on audio for two reasons. First, most game programming books ignor this subject entirely. Second, it is hard to find readable books or articles on the subject with the effects presented in plain language as this section did in all five articles. I notice that does not show the table of contents, so I do that here:

SECTION 1 GENERAL PROGRAMMING

Introduction 1.1  
 Game Object Component System 1.2  
 Closest- String Matching Algorithm 1.3  
 Implementing Lock-Free Algorithms 1.4  
 Computer Vision in Games using the OpenCV library 1.5  
 Using CPPUnit to implement unit testing 1.6  
 Visualizing Performance Data 1.7  
 Faster Loading with Access-Based File Reordering 1.8  
 Stay in the Game: Asset Hotloading for Fast Iteration 1.9  
 Geographic Grid Registration of Game Objects 1.10  
 Real-Time Continuous Profiling 1.11  
 Fingerprinting Prerelease Builds to Deter and Detect Piracy 1.12  
 The Dynamic Actor Layer

SECTION 2  
 MATHEMATICS AND PHYSICS

Introduction 2.1  
 Efficient Sequence Indexing 2.2  
 How to do floating point faster than hardware. 2.3  
 Solving linear systems by using the cross product 2.4  
 Real-Time Particle-Based Fluid Simulation 2.5  
 Exact Buoyancy for Polyhedra

SECTION 3  
 ARTIFICIAL INTELLIGENCE

Introduction 3.1  
 Designing a Multi-layer Pluggable AI Engine Framework 3.2  
 Smart Scene Load Management using Fuzzy Logic 3.3  
 Achieving coordination with autonomous NPC's 3.4  
 Using a Quantified Judgment Model for Engagement Analysis 3.5  
 Short-term memory using support vector machine (SVM) 3.6  
 Applying Model-Based Decision-Making Methods to Games 3.7  
 Constructing a Goal-Oriented Robot for UnrealTournament Using Fuzzy Sensors, Finite-state Behaviors and Behavior Networks

SECTION 4  
 SCRIPTING SYSTEMS

Introduction 4.1  
 A Script language survey 4.2  
 Managing High-Level Scripts Execution within 4.3  
 Automatic Function Binding for Game Scripting and Networking 4.4  
 Programming advanced control mechanisms with Lua coroutines 4.5  
 Binding C/C++ objects to Lua

SECTION 5  
 GRAPHICS

Introduction 5.1  
 Automatic Realistic Idle Motion Synthesis for Interactive Characters" 5.2  
 GPU Terrain Rendering 5.3  
 Rendering Road Signs Sharply 5.4  
 Interactive Fluid Dynamics and Rendering on the GPU 5.5  
 Spatial Partitioning using an Adaptive Binary Tree 5.6  
 Enhanced Object Culling with (Almost) Oriented Bounding Boxes 5.7  
 Fast per-pixel lighted scenes with a high number of lights 5.8  
 Efficient Sky Rendering Techniques 5.9  
 High Dynamic Range Rendering using OpenGL Frame Buffer Objects 5.10  
 Skin Splitting for Optimal Skin Rendering

SECTION 6  
 AUDIO

Introduction 6.1  
 Real- Time Sound Generation From Deformable Meshes 6.2  
 Volume Control Through Chained Busses 6.3  
 Faking Real-time DSP Effects 6.4  
 A lightweight oscillator for ambient sound generation 6.5  
 Cheap Doppler effect

SECTION 7  
 NETWORKING AND MULTIPLAYER

Introduction 7.1  
 Complex High-Level Systems 7.2  
 Reliable Peer-to-Peer Gaming Connection Penetrating Firewall and NAT 7.3  
 Massively Multiplayer Online Prototype (MMOP): Utilizing Second Life for Game Concept Prototyping 7.4  
 Dynamically Adaptive Streaming of 3D Data for Animated Characters 7.5  
 Generating globally unique identifiers for game objects

About the CD-ROM

Index

0 of 0 people found the following review helpful. watch this trend

By W Boudville

One interesting feature of this book on game programming is how graphics does NOT dominate the discussion. Most books on the subject are chock-a-block with every graphics tip you could imagine. But Dickheiser does not write for the novice. He has gathered a set of papers (chapters) that try to delineate the research boundary. It is a measure of the rising maturity of the field, and the complexity of many games, that graphics is relatively deprecated. Though I hasten to add that there are still the pretty colour plates of renderings, and several chapters on these methods. Why is the book symptomatic of the changing trends? In part because graphical methods are starting to saturate. The techniques are now good enough for near-photorealistic impressions. If you look carefully at the non-graphics chapters, you see that they tackle non-trivial problems arising from evermore intricate games. One instance is the use of AI-style decision making methods for how a game proceeds. When well coded, it can give behaviour of characters that is more intelligent, hence yielding a more interesting game. Another chapter looks at the learning process in AI, where a player might change strategies. Here, the research use of Support Vector Machines is suggested as apropos for commercial game development.

Professionally, you should pay attention to the book for the trend it exemplifies. You probably got into game programming because you were attracted by the graphics. But as graphics methods peak, the value added aspect of games is starting to shift towards the story line and how elaborate is the game logic.

13 of 21 people found the following review helpful. Programming Tips from the Pros

By John Matlock

Intended for the intermediate to advanced game programmer, this is a book that will literally have something for everyone. It contains fifty-two 'gems' that is, articles on how to some particular aspect of game programming. Sophisticated game programming is probably the most complex programming there is. And this book goes into some pretty sophisticated programming details. For

instance: 1.2 Using Multicore Processors - New software that facilitates programming to make the most effective use of dual/triple core CPUs. 2.5 - Exact Buoyancy for Polyhedra - Your game character has built a raft. How does it look like it's floating realistically? 5.9 - Practical Sky Rendering - The sky is so simple, unless, of course you want it to look real. ==There is no question that the main driving force for faster computers is to make for more realistic gaming. And the faster computers allow more sophisticated programming to be done to make the grass wave realistically. This is probably not a book you're going to sit down and read from cover to cover. You'll probably scan through and read the ones that are applicable to just what you need to do next. But then in a couple of weeks, in a couple of months....

Welcome to the sixth volume of the Game Programming Gems series. With team sizes constantly expanding, developers are finding themselves pushed to become increasingly specialized. This makes it important to have cutting-edge, ready-to-use material in your specialization, as well as resources that you can call upon if you need to work outside your area of expertise. Volume 6 has been explicitly designed with your current challenges in mind. With the new generation of machines, players expect higher-fidelity models and animations, fancier physics and graphics effects, and more intelligent AI. All of these capabilities require robust teams and longer schedules, so to help ensure that your games are still delivered on time and on budget, you'll find 50 all new articles written by experts in game technology from many different backgrounds and over twenty countries. This volume also includes a brand new section dedicated to Scripting and Data-Driven Systems, which is focused entirely on the growing trend of removing the programmer from the data-tweaking loop. So, whether you're a new game programmer starting out on this exciting path, a grizzled, industry veteran, or a most welcome visitor, we hope that you will find inspiration, insight, and at least one or two true gems to use in your world!

Foreword Preface About the Cover Image Contributor Bios SECTION 1 GENERAL PROGRAMMING Introduction 1.1 Lock-Free Algorithms 1.2 Utilizing Multicore Processors with OpenMP 1.3 Computer Vision in Games Using the OpenCV Library 1.4 Geographic Grid Registration of Game Objects 1.5 BSP Techniques 1.6 Closest-String Matching Algorithm 1.7 Using CppUnit To Implement Unit Testing 1.8 Fingerprinting Pre-Release Builds To Deter and Detect Piracy 1.9 Faster File Loading with Access-Based File Reordering 1.10 Stay in the Game: Asset Hotloading for Fast Iteration SECTION 2 MATHEMATICS AND PHYSICS Introduction 2.1 Floating-Point Tricks 2.2 GPU Computation in Projective Space Using Homogeneous Coordinates 2.3 Solving Systems of Linear Equations Using the Cross Product 2.4 Efficient Sequence Indexing for Game Development 2.5 Exact Buoyancy for Polyhedra 2.6 Real-Time Particle-Based Fluid Simulation with Rigid Body Interaction SECTION 3 ARTIFICIAL INTELLIGENCE Introduction 3.1 Applying Model-Based Decision-Making Methods to Games: Applying the Locust AI Engine to Quake III 3.2 Achieving Coordination with Autonomous NPCs 3.3 Behavior-Based Robotic Architectures for Games 3.4 Constructing a Goal-Oriented Robot for Unreal Tournament Using Fuzzy Sensors, Finite-State Machines, and Behavior Networks 3.5 A Goal-Oriented Unreal Bot: Building a Game Agent with Goal-Oriented Behavior and Simple Personality Using Extended Behavior Networks 3.6 Short-Term Memory Modeling Using a Support Vector Machine 3.7 Using the Quantified Judgment Model for Engagement Analysis 3.8 Designing a Multilayer, Pluggable AI Engine 3.9 A Fuzzy-Control Approach to Managing Scene Complexity SECTION 4 SCRIPTING AND DATA-DRIVEN SYSTEMS Introduction 4.1 Scripting Language Survey 4.2 Binding C/C++ Objects to Lua 4.3 Programming Advanced Control Mechanisms with Lua Coroutines 4.4 Managing High-Level Script Execution Within Multithreaded Environments 4.5 Exposing Actor Properties Using Nonintrusive Proxies 4.6 Game Object Component System SECTION 5 GRAPHICS Introduction 5.1 Synthesis of Realistic Idle Motion for Interactive Characters 5.2 Spatial Partitioning Using an Adaptive Binary Tree 5.3 Enhanced Object Culling with (Almost) Oriented Bounding Boxes 5.4 Skin Splitting for Optimal Rendering 5.5 GPU Terrain Rendering 5.6 Interactive Fluid Dynamics and Rendering on the GPU 5.7 Fast Per-Pixel Lighting with Many Lights 5.8 Rendering Road Signs Sharply 5.9 Practical Sky Rendering for Games 5.10 High Dynamic Range Rendering Using OpenGL Frame Buffer Objects SECTION 6 AUDIO Introduction 6.1 Real-Time Sound Generation from Deformable Meshes 6.2 A Lightweight Generator for Real-Time Sound Effects 6.3 Real-Time Mixing Busses 6.4 Potentially Audible Sets 6.5 A Cheap Doppler Effect 6.6 Faking Real-Time DSP Effects SECTION 7 NETWORK AND MULTIPLAYER Introduction 7.1 Dynamically Adaptive Streaming of 3D Data for Animated Characters 7.2 Complex Systems Based High-Level Architecture for Massively Multiplayer Games 7.3 Generating Globally Unique Identifiers for Game Objects 7.4 Massively Multiplayer Online Prototype Utilizing Second Life for Game Concept Prototyping 7.5 Reliable Peer-to-Peer Gaming Connections Penetrating NAT About the CD-ROM Index About the Author Michael Dickheiser (Raleigh, NC) is a Software Engineer with over nine years experience in team-oriented projects within the computer game industry. He has been involved in all stages of development from conceptual design, technical design and documentation, to implementation, debugging, and formal testing. Dickheiser was most recently a Senior Engineer at Red Storm Entertainment and he is now Senior Computer Scientist at Applied Research Associates.