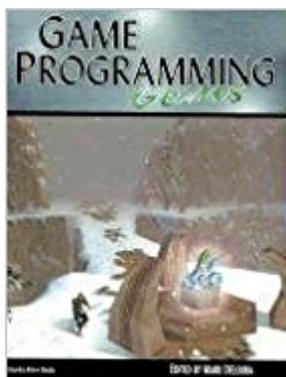The book was found

Game Programming Gems (Game Programming Gems (W/CD))

Synopsis

For the countless tasks involved in creating a game engine there are an equal number of possible solutions. But instead of spending hours and hours trying to develop your own answers, now you can find out how the pros do it! Game Programming Gems is a hands-on, comprehensive resource packed with a variety of game programming algorithms written by experts from the game industry and edited by Mark DeLoura, former software engineering lead for Nintendo of America, Inc. and now the newly appointed editor-in-chief of Game Developer magazine. From animation and artificial intelligence to Z-buffering, lighting calculations, weather effects, curved surfaces, mutliple layer Internet gaming, to music and sound effects, all of the major techniques needed to develop a competitive game engine are covered. Game Programming Gems is written in a style accessible to individuals with a range of expertise levels. All of the source code for each algorithm is included and can be used by advanced programmers immediately. For aspiring programmers, there is a detailed tutorial to work through before attempting the code, and suggestions for possible modifications and optimizations are included as well.

Book Information

Series: Game Programming Gems (W/CD)

Hardcover: 600 pages

Publisher: Charles River Media; Book & CD edition (August 2000)

Language: English

ISBN-10: 1584500492

ISBN-13: 978-1584500490

Product Dimensions: 9.5 x 7.3 x 1.7 inches

Shipping Weight: 2.9 pounds

Average Customer Review: 4.4 out of 5 stars   55 customer reviews

Best Sellers Rank: #672,151 in Books (See Top 100 in Books)   #183 inÃ Â Books > Arts & Photography > Other Media > Video Games   #424 inÃ Â Books > Computers & Technology > Games & Strategy Guides > Game Programming   #1136 inÃ Â Books > Computers & Technology > Programming > Graphics & Multimedia

Customer Reviews

Aimed at the working (or aspiring) Visual C/C++ game programmer, Game Programming Gems contains over 60 programming tips that have been gathered from more than 40 working game gurus. It you want to build your own games or are interested simply in how games work, this text

provides an intriguing glimpse into how the pros create state-of-the-art 3-D animation. The guiding principle in this book is to publish the best available tips for game programming; most of these fit into 10 pages or fewer. But don't let the efficient presentation fool you; almost every one of these tips will be invaluable to any serious game developer. Early sections concentrate on techniques for creating more maintainable, faster code. A guide to using scripts for data-driven game modules and techniques teaches you better resource management (like using handles). A quick-start tutorial to the Standard Template Library (STL) will help you learn how to use these fast collection classes in your code right away. Several contributors show off strategies for better game debugging and profiling--there's even a set of classes that can provide onscreen feedback during testing. The mathematical underpinnings that are required to do leading-edge 3-D graphics processing--including the use of quaternions, instead of matrices, in certain calculations--also are discussed. (One section looks at simulating water surfaces.) And artificial intelligence (AI) techniques for games--like Finite State Machines (FSMs), fuzzy logic, and neural networks--are explained. (The extremely cool flocking algorithms, which will let you add the behavior of birds or fish to your next game level, are especially appealing.) Over 20 techniques for doing work with polygons (a staple of representing 3-D virtual worlds) are laid out, too. You'll learn a variety of important concepts, such as collision detection, working with key frames, better skinning for character animation, and realistic terrain generation (including fractals). A discussion of pixel effects, with some cutting-edge ways to add more realistic lighting and shadows to your games, closes the discussion. (One of the more exciting sections shows you how to simulate glass objects within 3-D scenes.) With its leading-edge material on the algorithms that are used by the competition, Game Programming Gems will be a virtual must-read for anyone who works in the game industry. With code samples that are geared to OpenGL and that should run on both Windows and Linux, this book will help developers hone their game-programming skills. --Richard Dragan Topics covered: Tips and strategies for game developers Data-driven design and scripting languages Object-oriented design primer Using Visual C++ templates for faster math calculations Resource-management techniques (Singleton patterns, resource handles, and tips for fast data loads) The C++ Standard Template Library (STL) for games Bit arrays Network protocols for online games Using asserts and profiling for games Random numbers Interpolation methods Equations for rigid body motion Using polynomial approximations for trig functions Implicit Euler integration Wavelets Simulating water surfaces Quaternion vs. matrix calculations Artificial-intelligence (AI) techniques for gamers Sending messages Finite State Machines (FSMs) Game trees Pathing strategies (including A* and 3-D pathing solutions) Flocking algorithms Introduction to fuzzy logic

and neural networks Techniques for faster graphics with polygons (and 3-D fundamentals) Loading vertices faster into OpenGL The vector camera Camera-control strategies 3-D collision detection Multiresolution maps Distance calculationObject occlusion Working with octrees Interpolating between 3-D keyframes Skinning techniques Terrain-generation algorithms (including fractals) 2-D lens flare 2-D sprite effects with 3-D hardware Techniques for more realistic lighting Shadows and texturing Simulating glass and liquids in games

Key Features: * A must-have for every game programmer's library! * Written by Game Programming Experts and edited by Nintendo's Mark DeLoura * Comprehensive coverage of all major techniques used in game development * CD ROM is packed with the source code in C & C++ completely portable to Windows and Linux, and all graphics displays use the popular Open GL language

The book is comprised of five sections, each containing many short articles, usually written by different authors, and each article contains references. The book assumes that you are familiar with c++, or c at the minimum, and the examples are all written in these languages. Also, the graphics library supported by the book is OpenGL, not DirectX. If you are new to c or c++ and haven't done any game programming, then this book will be too advanced for you. The code is suitable for both Windows and Linux platforms.The first section of the book discusses object-oriented design and often refers to "design patterns". Since I hadn't heard of them, and programmed my own way, there was much to learn. This is where the book's references help. If you aren't an expert at c++ programming, I suggest you read the references, namely "Effective c++", "More Effective c++" (both by Scott Meyers), and also the book on design patterns mentioned: "Design Patterns" by Gamma et. al. To make the most of the STL section, you could do with a book on STL also.The chapter on bit-arrays is especially good for compression and decompression code, preventing you from having to perform tricky bit operations. The profiling section is good because it shows how to profile with your own code, rather than relying on having a compiler to profile for you. That's just two example chapters in this section.The second section of the book is about Maths. To me, the articles are too complex given how short they are. To understand them, you need university-level Maths and understand summing notation, vectors, matrices, differentiation, and integrals. I've only done First Year university Maths, and found myself out of my depth in this section.The third section is about Artificial Intelligence. This section covers finite state machines (useful in any game), game trees (used by the computer opponent to defeat the human player in games such as chess and checkers), as well as search techniques, how to make enemies flock, and fuzzy logic.The fourth

section is about polygonal techniques, which are mainly used for 3D games. The book's code for 3D games is written with OpenGL in mind. I am mostly interested in 2D game design, so I haven't read all the chapters of this section. But there's an interesting chapter that shows you how to generate smooth curves given a set of control points. This is great for making an enemy move along a curve rather than in a straight line, so you can apply it to 2D game design.The final section has a chapter on using 3D hardware for 2D sprite effects (with OpenGL). This is exactly what I needed to know. It was my favourite article because it solved my immediate problem, the answer to which I couldn't find after exhaustive web searches. This book isn't just for 3D game creation; it shows how to use OpenGL for 2D games. This way you can get the benefits of sprite scaling and rotation in a 2D game.The accompanying CD has the full source code to the articles, which is required because many chapters only contain code snippets with the text.Overall, this is a great book. It is big and comprehensive. It's better than a single-author book because you're not stuck in his particular mindset. Plus you can pick up the book and quickly read a chapter, and it's usually not necessary to read the chapters in order. The disadvantage is that some of the chapters may not interest you, but there's plenty that will interest you, as the book has a large number of articles, and the book size is 600 pages. The book is well-balanced in that it doesn't contain too much theory, nor too little, and it contains practical examples and utilities that you can readily use.Anyone with c++ knowledge and an interest in programming games should buy this book. If your c++ skills aren't too sharp, you will want to buy some of the referenced books.I look forward to reading the other books in the series.

This book is hands-down the best book yet published on game programming. I have yet to find any other book that begins to approach the excellence of Game Programming Gems in terms of the breadth and depth of the subjects covered.GPG will serve as an excellent introduction to a broad variety of game programming techniques for those new to the industry, and an invaluable desk reference and for more experienced game developers. As a 7-year industry veteran, I can't count the number of times the techniques in this book would have proven useful in the past.Of particular interest are Steve Rabin's excellent chapters on the A* algorithm, the cornerstone of (most) pathfinding in computer games. These chapters go far beyond the explanation of the algorithm itself and serve up a host of rare and valuable insights for getting the most out of your pathfinding in an actual game environment.I have no doubt that this book will have a significant impact on the state of the art in the game development community, and one can only hope that this book is only a hint of what's to come.

Got what I needed. I heard it was a good book. I've only skimmed through it. I like it so far.

The book is good. The fact that the CD was missing for the book when it arrived at my door, and that the source code for the book is unavailable for honest download online is very irritating. The book does describe the algorithms at a high level and it includes example source code in the text. However, the example implementation of the source code will not help you very much without the actual source code, because not all data structures are very well described.As an example: pages 287-304 describes a simple method of navigating in a 3D space. Some of the methods for the NavigationMesh and NavigationCell classes are defined in plain text, but the classes themselves are not defined (in the text). Without the source code... it is a lot better than nothing.Another example: Chapter 2.6 describes the process of solving the wave equation in 2D. It is no substitute for a full book on how to solve partial differential equations using the finite difference method. But it does solve the wave equation using it! He also gives the analytical solution to the wave equation (he doesn't explain where that comes from, but that's fine). He omits a *very* important detail regarding the courant-friedrich lewy condition (the stability condition) that will have the unfortunate reader who is unfamiliar with PDEs tearing their hair out of their skull if they attempt to implement the solution themselves [pick the wrong combination of dt and h, and the uninitiated reader could be left howling in frustration for hours wondering why the stupid simulation they wrote keeps generating garbage for numbers!]. And the author of this chapter could have fixed this with a more detailed explanation of how solving these PDEs actually *really* works and why they work! To his credit, he includes some pretty good references at the end, the sections written by this fellow were very well written, the supporting figures are very good, his descriptions are good! He just left out some important details that are absolutely necessary to have if you really want to implement the simulation (like the where the CFL condition comes from, and how the boundary conditions work).This book will help keep you from wandering in the dark and being forced to figure out all of your own novel algorithms for solving problems that occur in writing a game engine. The text itself provides a theoretical and algorithmic overview of the issues of solving problems with game engines. It will get you thinking in the right direction. But the implementation details are (presumably) left in the CD (which I cannot look at, since my book didn't have a CD shipped with it). And the CD has source code for linux and windows. (It uses OpenGL and GLUT for much of it's source code, so it should be possible to get the code to compile on a modern macintosh, with some effort).

Love this book. Very clear writing style!

A must-have book for game programmers.

Heard great things about this book. Have not read the whole thing, but so far it looks good so far.

One of the must have books if you are interested in game development.

Game Programming Gems (Game Programming Gems (W/CD)) Python Programming: Python Programming for Beginners, Python Programming for Intermediates, Python Programming for Advanced C++: The Ultimate Crash Course to Learning the Basics of C++ (C programming, C++ in easy steps, C++ programming, Start coding today) (CSS,C Programming, ... Programming,PHP, Coding, Java Book 1) C++ and Python Programming: 2 Manuscript Bundle: Introductory Beginners Guide to Learn C++ Programming and Python Programming C++ and Python Programming 2 Bundle Manuscript. Introductory Beginners Guide to Learn C++ Programming and Python Programming Python Programming: The Complete Step By Step Guide to Master Python Programming and Start Coding Today! (Computer Programming Book 4) Game of Thrones: 100 Question Trivia Game For True Fans (Epic Fantasy Series, Game of Thrones Books, Game of Thrones, Fantasy Books) (Epic Fantasy, Fantasy ... TV, TV Guide, Game of Thrones Book) Game Programming for Teens (Premier Press Game Development) Programming Game AI By Example (Wordware Game Developers Library) Python Programming Advanced: A Complete Guide on Python Programming for Advanced Users PYTHON: LEARN PYTHON in A Day and MASTER IT WELL. The Only Essential Book You Need To Start Programming in Python Now. Hands On Challenges INCLUDED! (Programming for Beginners, Python) Python Programming Guide + SQL Guide - Learn to be an EXPERT in a DAY!: Box Set Guide (Python Programming, SQL) The Complete Software Developer's Career Guide: How to Learn Your Next Programming Language, Ace Your Programming Interview, and Land The Coding Job Of Your Dreams Programming with MicroPython: Embedded Programming with Microcontrollers and Python CNC 50 Hour Programming Course: For lathes, ISO Standard functions, Siemens fixed cycles, parametric programming, methods of use Assessment, Evaluation, and Programming System for Infants and Children (AEPSÃ Â®), Second Edition, Curriculum for Three to Six Years (AEPS: Assessment, Evalutaion, and Programming System (Unnumbered)) Introduction to Programming with Greenfoot: Object-Oriented Programming in Java with Games and Simulations (2nd Edition) Data Analytics and Python Programming: 2 Bundle Manuscript: Beginners Guide to Learn Data Analytics, Predictive

Analytics and Data Science with Python Programming ESP8266: Programming NodeMCU Using Arduino IDE - Get Started With ESP8266 (Internet Of Things, IOT, Projects In Internet Of Things, Internet Of Things for Beginners, NodeMCU Programming, ESP8266) Java: 2017 Ultimate Beginners Guide to Learn Java Programming (java for dummies, java apps, java for beginners, java apps, hacking, hacking exposed) ... Programming, Developers, Coding, CSS, PHP)