

Foreword by John Carmack of id Software

Michael Abrash's
GRAPHICS
PROGRAMMING
Black Book
SPECIAL EDITION

Michael Abrash



What's on the CD-ROM

The companion CD-ROM includes all of the source code published as numbered listings in the text of the book, plus compiled executables of many of the demos. In addition, you'll find the following extras on the CD:

- The classic Zen Timer code profiling tool, in both executable and source code format.
- Exclusive! The text of Michael's long out of print 1989 cult classic *Zen of Assembly Language*, plus scans of all 100+ technical figures.
- Several significant essays from Michael's ongoing work in game development, appearing for the first time in book form.

See the *readme* file for acknowledgments, descriptions, copyrights, installation, limitations, and other important information.

Requirements

Software A Microsoft operating system, a MASM-compatible assembler, and a Microsoft-compatible C compiler are required to rebuild much of the code. Some code is C-only. Much includes assembly modules. Some code requires Windows 95 or NT.

Hardware Platform: An Intel PC.
Note that some code is processor-specific. To run all code you must have at least a Pentium processor.

Michael Abrash's
GRAPHICS
PROGRAMMING
Black Book
SPECIAL EDITION

Michael Abrash's
GRAPHICS
PROGRAMMING
Black Book
SPECIAL EDITION

Michael Abrash

 **COROLIS GROUP BOOKS**

an International Thomson Publishing company **ITP**[®]

Albany, NY • Belmont, CA • Bonn • Boston • Cincinnati • Detroit • Johannesburg • London
Madrid • Melbourne • Mexico City • New York • Paris • Singapore • Tokyo • Toronto • Washington

Publisher	<i>Keith Weiskamp</i>
Project Editor	<i>Denise Constantine</i>
Production Project Coordinator	<i>Kim Eoff</i>
Composer	<i>Rob Maubar</i>
Cover Artist and Cover Design	<i>Anthony Stock</i>
Proofreaders	<i>Jeff Kellum and Meredith Brittain</i>
Indexer	<i>Caroline Parks</i>
CD-ROM Development	<i>Robert Clarfield</i>

Michael Abrash's Graphics Programming Black Book, Special Edition

1-57610-174-6

Copyright © 1997 by The Coriolis Group, Inc.

All rights reserved. This book may not be duplicated in any way without the express written consent of the publisher, except in the form of brief excerpts or quotations for the purposes of review. The information contained herein is for the personal use of the reader and may not be incorporated in any commercial programs, other books, databases, or any kind of software without written consent of the publisher. Making copies of this book or any portion for any purpose other than your own is a violation of United States copyright laws.

Limits of Liability and Disclaimer of Warranty

The author and publisher of this book have used their best efforts in preparing the book and the programs contained in it. These efforts include the development, research, and testing of the theories and programs to determine their effectiveness. The author and publisher make no warranty of any kind, expressed or implied, with regard to these programs or the documentation contained in this book.

The author and publisher shall not be liable in the event of incidental or consequential damages in connection with, or arising out of, the furnishing, performance, or use of the programs, associated instructions, and/or claims of productivity gains.

Trademarks

Trademarked names appear throughout this book. Rather than list the names and entities that own the trademarks or insert a trademark symbol with each mention of the trademarked name, the publisher states that it is using the names for editorial purposes only and to the benefit of the trademark owner, with no intention of infringing upon that trademark.

The Coriolis Group, Inc.
 An International Thomson Publishing Company
 14455 N. Hayden Road, Suite 220
 Scottsdale, Arizona 85260

602/483-0192
 FAX 602/483-0193
<http://www.coriolis.com>

Printed in the United States of America
 10 9 8 7 6 5 4 3

Acknowledgments

Because this book was written over many years, in many different settings, an unusually large number of people have played a part in making this book possible. First and foremost, thanks (yet again) to Jeff Duntemann for getting this book started, doing the dirty work, and keeping things on track and everyone's spirits up. Thanks to Dan Illowsky for not only contributing ideas and encouragement, but also getting me started writing articles long ago, when I lacked the confidence to do it on my own—and for teaching me how to handle the business end of things. Thanks to Will Fastie for giving me my first crack at writing for a large audience in the long-gone but still-missed *PC Tech Journal*, and for showing me how much fun it could be in his even longer-vanished but genuinely terrific column in *Creative Computing* (the most enjoyable single column I have ever read in a computer magazine; I used to haunt the mailbox around the beginning of the month just to see what Will had to say). Thanks to Robert Keller, Erin O'Connor, Liz Oakley, Steve Baker, and the rest of the cast of thousands that made *PJa* a uniquely fun magazine—especially Erin, who did more than anyone to teach me the proper use of English language. (To this day, Erin will still patiently explain to me when one should use “that” and when one should use “which” even though eight years of instruction on this and related topics have left no discernible imprint on my brain.) Thanks to Jon Erickson, Tami Zemel, Monica Berg, and the rest of the *DDJ* crew for excellent, professional editing, and for just being great people. Thanks to the Coriolis gang for their tireless hard work: Jeff Duntemann and Keith Weiskamp on the editorial and publishing side, and Brad Grannis, Rob Mauhar, and Kim Eoff who handled art, design, and layout. Thanks to Jim Mischel who did a terrific job testing code for the book and putting the code disk together. Thanks to Jack Tseng, for teaching me a lot about graphics hardware, and even more about how much difference hard work can make. Thanks to John Cockerham, David Stafford, Terje Mathisen, the BitMan, Chris Hecker, Jim Mackraz, Melvin Lafitte, John Navas, Phil Coleman, Anton Truenfels, John Carmack, John Miles, John Bridges, Jim Kent, Hal Hardenberg, Dave Miller, Steve Levy, Jack Davis, Duane Strong, Daev Rohr, Bill Weber, Dan Gochnauer, Patrick Milligan, Tom Wilson, the people in the `ibm.pc/fast.code` topic on Bix, and all the rest of you who have been so generous with your ideas and suggestions. I've done my best to acknowledge contributors by name in this book, but if your name is omitted, my apologies, and consider yourself thanked; this book could not have happened without you. And, of course, thanks to Shay and Emily for their generous patience with my passion for writing and computers.

And, finally, thanks to the readers of my articles and to you, the reader of this book. You are, after all, the ultimate reason why I write, and I hope you learn as much and have as much fun reading this book as I did writing it!

Michael Abrash (mikeab@microsoft.com)
Bellevue, Washington, 1997

To Shay

Foreword

I got my start programming on Apple II computers at school, and almost all of my early work was on the Apple platform. After graduating, it quickly became obvious that I was going to have trouble paying my rent working in the Apple II market in the late eighties, so I was forced to make a very rapid move into the Intel PC environment.

What I was able to pick up over several years on the Apple, I needed to learn in the space of a few months on the PC.

The biggest benefit to me of actually making money as a programmer was the ability to buy all the books and magazines I wanted. I bought a lot. I was in territory that I knew almost nothing about, so I read *everything* that I could get my hands on. Feature articles, editorials, even advertisements held information for me to assimilate.

John Romero clued me in early to the articles by Michael Abrash. The good stuff. Graphics hardware. Code optimization. Knowledge and wisdom for the aspiring developer. They were even fun to read. For a long time, my personal quest was to find a copy of Michael's first book, *Zen of Assembly Language*. I looked in every bookstore I visited, but I never did find it. I made do with the articles I could dig up.

I learned the dark secrets of the EGA video controller there, and developed a few neat tricks of my own. Some of those tricks became the basis for the Commander Keen series of games, which launched id Software.

A year or two later, after Wolfenstein-3D, I bumped into Michael (in a virtual sense) for the first time. I was looking around on M&T Online, a BBS run by the Dr. Dobb's publishers before the Internet explosion, when I saw some posts from the man himself. We traded email, and for a couple months we played tag-team gurus on the graphics forum before Doom's development took over my life.

A friend of Michael's at his new job put us back in touch with each other after Doom began to make its impact, and I finally got a chance to meet up with him in person.

I talked myself hoarse that day, explaining all the ins and outs of Doom to Michael and an interested group of his coworkers. Every few days afterwards, I would get an email from Michael asking for an elaboration on one of my points, or discussing an aspect of the future of graphics.

Eventually, I popped the question—I offered him a job at id. “Just think: no reporting to anyone, an opportunity to code all day, starting with a clean sheet of paper. A chance to do *the right thing* as a programmer.” It didn’t work. I kept at it though, and about a year later I finally convinced him to come down and take a look at id. I was working on Quake.

Going from Doom to Quake was a tremendous step. I knew where I wanted to end up, but I wasn’t at all clear what the steps were to get there. I was trying a huge number of approaches, and even the failures were teaching me a lot. My enthusiasm must have been contagious, because he took the job.

Much heroic programming ensued. Several hundred thousand lines of code were written. And rewritten. And rewritten. And rewritten.

In hindsight, I have plenty of regrets about various aspects of Quake, but it is a rare person that doesn’t freely acknowledge the technical triumph of it. We nailed it. Sure, a year from now I will have probably found a new perspective that will make me cringe at the clunkiness of some part of Quake, but at the moment it still looks pretty damn good to me.

I was very happy to have Michael describe much of the Quake technology in his ongoing magazine articles. We learned a lot, and I hope we managed to teach a bit.

When a non-programmer hears about Michael’s articles or the source code I have released, I usually get a stunned “WTF would you do that for???” look. They don’t get it.

Programming is not a zero-sum game. Teaching something to a fellow programmer doesn’t take it away from you. I’m happy to share what I can, because I’m in it for the love of programming. The Ferraris are just gravy, honest!

This book contains many of the original articles that helped launch my programming career. I hope my contribution to the contents of the later articles can provide similar stepping stones for others.

—John Carmack
id Software