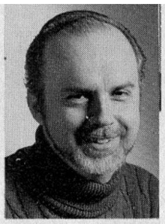


Objects and Graphics

Reviewed by Walter Zintz

Object-oriented software and computer graphics are surprisingly relevant to each other. Objects are an efficient way to implement the inherently parallel computations of computer graphics, and graphics programs are an easy way to spotlight the basics of object orientation.

This month I review books in both areas. The relationship between the two is mentioned lightly if at all, so you'll have to make the connection yourself, but the fundamentals of both fields are covered nicely.

Photorealism and Ray Tracing in C

By Christoper D. Watkins, Stephen B. Coy, [redacted]
Published by M&T Books
482 pages, paperbound with 5 1/4-inch diskettes, $44.95
Technical Level: Intermediate to Advanced
Understandability: Crystal Clear
Summary: All you need to get started in ultra-realistic computer imaging

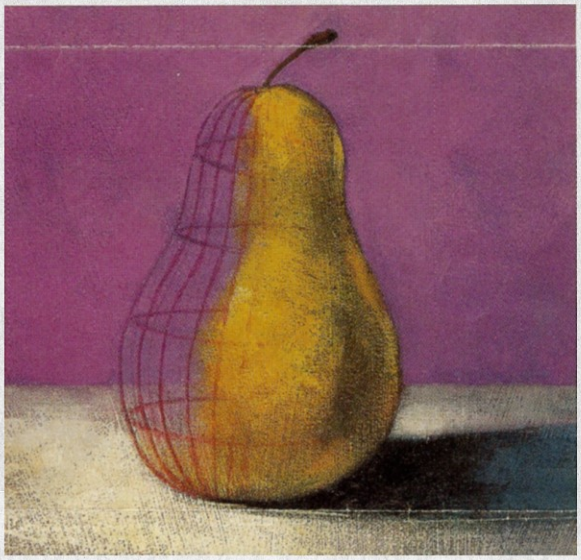
Shading techniques that allow you to create semi-realistic computer images of 3-D scenes exist in spades. But whenever you see an image that looks as good as–or better than–a professional photograph, you can bet it was done by tracing rays.

"Ray tracing" means following the path of theoretical light rays, from the imagined viewer's eye back through the computer screen, bouncing off one or more postulated objects in the virtual scene, and finally back to an imagined light source. Half a dozen years ago, ray tracing was a business for supercomputers, but dramatic improvements in both computer power and computational algorithms make ray tracing practical today on many desktop systems.

This book serves as a smooth introduction for programmers who may not know a single thing about tracing rays. It starts with a quick review of the C language, then explains the basic graphics code modules used in the book and the advanced math modules that underlie them. The second part tells you what ray tracing is all about, ways to attack aliasing problems, how programs can decide whether a ray will strike an object, the basics of light as it bounces off or passes through objects it encounters, and how the ray-tracing code presented works. The next third of the book tells how to create databases of scene information that the ray tracer can readily use, and explains a few advanced topics in image processing software and hardware.

The programs used in the book are pretty easy to understand because they're simplified. They don't deal with matters like animation and the complexities of intra-scene light diffusion, although the last chapter touches briefly on these issues and gives a few sources for further information. All the code is on the diskettes as well as in the text. Unfortunately, the code is set up for DOS, but the dependencies are mostly in the output drivers.

The writing is surprisingly seamless for a multiauthor book. It is directed toward readers who are primarily programmers and are just passably familiar with college-level math and standard methods in computer graphics. The style is appropriate to the level, neither above nor below the intended audience's heads. To keep enthusiasm up, there are 16 full-color pages of gorgeous ray-traced images.



Object-Oriented Programming

By Peter Coad and Jill Nicola
Published by Yourdon Press (Prentice Hall)
582 pages, hardbound with 3 1/2-inch diskette, $39.00
Technical Level: Beginner to Expert
Understandability: Almost Crystal Clear
Summary: Getting really inside object orientation

Can this book be serious? It starts out telling (presumably adult) programmers to make up and recite little ditties like "I am a counter; I know my value and my reset value; I know how to increment, decrement and reset myself." It espouses principles with names like "nice threads," "pay up," "heart and soul," "mail dominance," and "me, myself, and I."

Yes, this book is very serious. The aspects I've cited are among its most important features. And the result is the best book I've seen yet for helping software people who are just confused by all the other books on object orientation.

Take the ditties. They sound like what we all left behind in kindergarten, but they're what we need to say in order to see what we're doing from the right perspective.

Belying its title, this book deals with all three aspects of object orientation:

CHRIS McALLISTER