
Foreword



Excellent! I am glad that you are reading this book. You might want to skip straight ahead to the good stuff, but as long as you are here...

Computer graphics is a fascinating and fast-changing field that didn't even exist when I was born. I was attracted to it because it is a field with a unique mix of engineering and artistry. In the computer graphics industry, people with engineering skills design graphics software and hardware products that offer ever-increasing levels of performance and image quality. These products inspire people with artistic skills to use the resulting products to create amazing visual experiences that entertain, teach, or help others create or design. This in turn inspires the engineers to create even better hardware and software in order to improve the visual experiences created by artists. This symbiotic relationship between engineers and artists has never let up and has

resulted in photorealistic effects for movies and near-cinematic quality experiences for computer games.

You might be reading this book because of your interest in the computer graphics field. Perhaps you are an engineer looking to develop another tool for your toolbox of software development skills for computer graphics. Perhaps you are an artist who is interested in learning a little more about the bits and bytes of how computer graphics images are created. Perhaps you are that rare breed, an engineer/artist, and you have in your mind's eye a vision of what you want to create, and you need only to develop an understanding of this new medium in order to bring your vision to reality. If any of these are true, you have selected an excellent guide book to help you on your journey.

You are holding in your hands a book written by two people who share two passions. Mike Bailey and Steve Cunningham both love computer graphics, and they are absolutely passionate about teaching. This book allows them to combine both of these passions into a form that is sure to benefit you, the reader.

Actually, the word “passionate” understates the impact that Mike and Steve have had on computer graphics education. Mike is a “lifer” in the computer graphics industry. I met him some 15 years ago when we asked him to lead an effort to define industry-standard benchmarks for computer graphics systems (which he graciously agreed to do). He has been teaching or practicing computer graphics for almost 30 years now. He has won numerous awards as a professor of computer graphics. His dedication to educating people new to graphics is demonstrated by the fact that he annually prepares and delivers the “Introduction to Computer Graphics” tutorial at SIGGRAPH (ACM's Special Interest Group on Graphics).

Steve is a similarly dedicated, accomplished, and award-winning educator. He was a co-founder of the SIGGRAPH Education Committee and co-chaired this activity for many years. He served in countless leadership positions in the SIGGRAPH organization and for the SIGGRAPH conference itself (the largest, most prestigious, and longest-lived conference focusing on computer graphics). For his lifelong efforts, he was given the 2004 ACM SIGGRAPH Outstanding Service Award. His influence on the computer graphics industry is global, as witnessed by the fact that he was the first Eurographics Education Board chair and he has been named a Eurographics Fellow.

So it is certainly the case that these two authors can tell you a thing or two about computer graphics. But even more importantly, they can tell it to you in a way that you will understand and remember.

The topic of this book, writing shaders with the OpenGL Shading Language, is both important and timely. OpenGL and its companion shading

language are industry standards. This means that they are supported by a variety of hardware companies on a variety of operating environments. OpenGL and GLSL are available on Macs, PCs, and Linux systems; on workstations, towers, desktops, laptops, and handhelds. The goal of a standard is simple: to make it easy for you, the programmer, to deploy your code on a diverse range of products without requiring any changes to the source code. The resulting portability amortizes the cost of the software development by creating a bigger market for software products based on industry standards.

But the most important part of this book is that while it is teaching you how to write programmable shaders, it is also teaching and reinforcing the fundamentals of computer graphics. As a result, you will be able to easily adapt the lessons learned here to other shading languages and graphics paradigms. This is becoming increasingly important since the trend for graphics hardware is to offer more general programmability and less fixed functionality built into hardware. In other words, we are returning to the days where computer graphics innovation occurs in software. The knowledge and skills that you learn while reading this book can be adapted to the even more general graphics programming environments of the future.

At the end of each chapter in this book, you will find some exercises that will help develop your knowledge of graphics and programmable shading. In that spirit, here are the exercises that I would prescribe for you:

1. Read this book.
2. Use computer graphics and programmable shading to create beauty.
3. Share your creation and your knowledge with others.

Most importantly,

4. Have fun!

Randi Rost
December 31, 2008