About the Author

Christer Ericson is a senior principal programmer and the tools and technology lead at Sony Computer Entertainment America in Santa Monica. Before joining Sony in 1999, he was a senior programmer at Neversoft Entertainment. Christer received his Masters degree in computer science from Umeå University, Sweden, where he also lectured for several years before moving to the US in 1996. Christer has served on the advisory board for Full Sail's Game Design and Development degree program since 2002. His interests are varied, but he takes a particular interest in program optimization, a topic he has spoken on at the Game Developers Conference.

Contents

Preface

Together with a friend, I wrote my first computer game as a preteen, in 1978—the same year as Space Invaders was released. Written in BASIC, our game was a quiz game where you were asked questions about African animals. Compared to Space Invaders, our text-based game was primitive and not very exciting. Still, we were hooked, and it was not long until we were writing copies on our home computers, not only of Space Invaders but also of many other arcade games of that period, not to mention creating an endless number of original games of our own design. My then-hobby of writing games has today become my day job and games have evolved into a multi-billion dollar industry, which—for better or worse—virtually single-handedly drives the development of graphics hardware and fuels the need for increasingly more powerful CPUs.

Back then, one of the main challenges to writing an action game was dealing with *collision detection*: the problem of determining if an object had intersected another object or overlapped relevant background scenery. Since games were (primarily) 2D, collision detection involved determining overlap in screen space in efficient ways. Interestingly, even though computers today are over 1000 times faster, collision detection remains a key challenge. Today, game worlds are predominantly in 3D. They are of incredible complexity, containing tens if not hundreds of millions of polygons. Collision detection solutions now require sophisticated data structures and algorithms to deal with such large data sets, all of this taking place in real-time. Of course, games are not the only applications having to solve complex collision detection problems in real-time; other applications, such as CAD/CAM systems and 3D modeling programs must also address these problems.

The goal of this book is to provide efficient solutions for games and all other real-time applications to address their collision detection problems. To make this possible, this book provides an extensive coverage of the data structures and algorithms related to collision detection systems. Implementing collision detection systems also requires a good understanding of various mathematical concepts, which this book also focuses on. Special care has been taken to discuss only practical solutions, and code and pseudocode is provided to aid the implementation of the methods discussed in the book.

Overall, collision detection is a very large topic. Every chapter in this book could easily form the basis of a book each. As such, the coverage has been restricted to the most important areas and that provide a solid foundation for further exploration into this rich field.