

THOMSON

COURSE TECHNOLOGY

Professional • Technical • Reference

# BEGINNING GAME PROGRAMMING

SECOND EDITION

JONATHAN S. HARBOUR

FOREWORD BY  
BARRY E. CAUDILL  
EXECUTIVE PRODUCER, FIRAXIS GAMES





# BEGINNING GAME PROGRAMMING

**SECOND EDITION**

**JONATHAN S. HARBOUR**

**THOMSON**



**COURSE TECHNOLOGY**

Professional ■ Technical ■ Reference

© 2007 Thomson Course Technology, a division of Thomson Learning Inc. All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system without written permission from Thomson Course Technology PTR, except for the inclusion of brief quotations in a review.

The Thomson Course Technology PTR logo and related trade dress are trademarks of Thomson Course Technology, a division of Thomson Learning Inc., and may not be used without written permission.

Windows, DirectX, and Visual C++ are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Borland C++ and C++Builder are trademarks of Borland Software Corporation in the United States and other countries. Pro Motion is a copyright of Cosmigo GmbH. Anim8or is a copyright of Steve Glanville. Mappy is a copyright of Robin Burrows. Ghost in the Shell, Motoko Kusanagi, and Section 9 are copyrights of Shirow Masamune-Production I.G./KODANSHA.

All other trademarks are the property of their respective owners.

*Important:* Thomson Course Technology PTR cannot provide software support. Please contact the appropriate software manufacturer's technical support line or Web site for assistance.

Thomson Course Technology PTR and the author have attempted throughout this book to distinguish proprietary trademarks from descriptive terms by following the capitalization style used by the manufacturer.

Information contained in this book has been obtained by Thomson Course Technology PTR from sources believed to be reliable. However, because of the possibility of human or mechanical error by our sources, Thomson Course Technology PTR, or others, the Publisher does not guarantee the accuracy, adequacy, or completeness of any information and is not responsible for any errors or omissions or the results obtained from use of such information. Readers should be particularly aware of the fact that the Internet is an ever-changing entity. Some facts may have changed since this book went to press.

Educational facilities, companies, and organizations interested in multiple copies or licensing of this book should contact the Publisher for quantity discount information. Training manuals, CD-ROMs, and portions of this book are also available individually or can be tailored for specific needs.

ISBN-10: 1-59863-288-4

ISBN-13: 978-1-59863-288-0

eISBN-10: 1-59863-786-X

Library of Congress Catalog Card Number: 2006904402

Printed in the United States of America

07 08 09 10 11 PH 10 9 8 7 6 5 4 3 2 1

**THOMSON**  
  
**COURSE TECHNOLOGY**  
Professional ■ Technical ■ Reference

Thomson Course Technology PTR,  
a division of Thomson Learning Inc.  
25 Thomson Place  
Boston, MA 02210  
<http://www.courseptr.com>

**Publisher and General Manager,**  
**Thomson Course Technology PTR:**  
Stacy L. Hiquet

**Associate Director of Marketing:**  
Sarah O'Donnell

**Manager of Editorial Services:**  
Heather Talbot

**Marketing Manager:**  
Heather Hurley

**Senior Acquisitions Editor:**  
Emi Smith

**Marketing Coordinator:**  
Adena Flitt

**Project Editor:**  
Jenny Davidson

**Technical Reviewer:**  
Joshua R. Smith

**PTR Editorial Services Coordinator:**  
Erin Johnson

**Interior Layout Tech:**  
ICC Macmillan Inc.

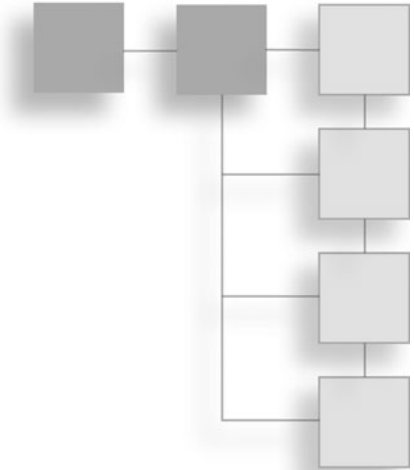
**Cover Designer:**  
Mike Tanamachi

**CD-ROM Producer:**  
Brandon Penticuff

**Indexer:**  
Kelly D. Henthorne

*For My Mother,  
Vicki Myrlene Harbour*

# FOREWORD



“I want to be a game designer, how do I get a job?” This is a question I field very often when I do interviews or talk to students. I’ve even been accosted by the parents of an apparently gifted teenager as I left the stage with my band. My usual answer is, “so what have you designed?” The vast majority of the time, I am given a long explanation about how the person has lots of great ideas, but is in need of a team to make them a reality. My response to this is to try to explain how everyone I work with has great ideas, but only a small percentage of them are designers.

I don’t mean to be harsh, but the reality is that there are no successful companies out there that will give someone off the street a development team for 18+ months and a multimillion dollar budget without some sort of proof of concept. What sets someone like Sid Meier (legendary game designer with whom I’m honored to work at Firaxis Games) apart is his ability to take an idea and make something fun out of it. Of course, Sid now gets large teams to do his projects, but he always starts the same way—a team of one cranking out prototypes cobbled together with whatever art and sound he can either dig up or create himself. It’s these rough proofs of concept that allow people uninvolved with the creation process to immediately see the fun in a given idea, and that’s what gets you a budget and a team. Every budding designer should take note and ask, “What would Sid do?”

That’s when a book like this is invaluable. I became acquainted with Jonathan a couple of years ago when I picked up the original version of this book at the bookstore at the Game Developer’s Conference. A programmer buddy of mine

helped me pick it out from among numerous similar books. He thought it was very well written and thought the emphasis on DirectX would be very applicable to what we do at Firaxis. Another buddy mentioned that he had read Jonathan's work on programming the Game Boy Advance and was very impressed. In my opinion, they gave me great advice and I enjoyed myself immensely while working through the book. While reading, I noticed that Jonathan was a big fan of our game, *Sid Meier's Civilization III*. I contacted him because I have worked on numerous *Civ* titles and we have kept in contact ever since.

The beauty of a book like this is that it takes away all of the excuses. It provides an excellent introduction to game programming. It takes you by the hand and walks you through the seemingly complex process of writing C code making use of DirectX. Before you know it, you'll have a fully usable framework for bringing your ideas to life. You are even provided with tools to create your own art and sound to help dress up the game. In other words, you will have all the tools you need to start making prototypes and prove that you are much more than just someone with great ideas. Believe me; taking this crucial next step will put you at the top of the heap of people looking for jobs in the industry. You will have the ability to stand out and that's vital when so many people are clamoring for work in game development.

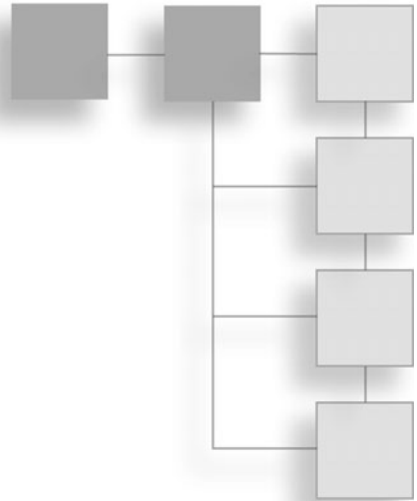
So, what would Sid do? Well, when he was prototyping *Sid Meier's Railroads!* last year, he wrote the entire prototype in C. He didn't have an artist (they were all busy on another title at the time), so he grabbed a 3D art program, made his own art, and threw it in the game—often using text labels to make sure players knew what things were in the game. He used audio files from previous Firaxis games and the Internet, and sprinkled them around to enhance the player's experience. He created something—in a fairly short amount of time—that showed our publisher and others just how much fun the game was going to be. And he did it on his own . . . just like the “old days” when he worked from his garage.

So what should you do? Well, if you want to get a job in the industry as a game designer or even if you just want to make a cool game to teach math to your daughter, you should buy this book. Jump in and work through the exercises and develop the beginnings of your own game library—Sid has some code he's used since the Commodore 64 days. Let your imagination run wild and then find ways to translate your ideas into something people can actually play. Whatever you do, just do *something*. It's the one true way to learn and develop as a designer and it is your ticket to finding game designer fulfillment and maybe even a job. And if Sid

wasn't Sid, and didn't already have all of those tools at his disposal, it just might be what he would do too.

**Barry E. Caudill**  
**Executive Producer**  
**Firaxis Games**  
**2K Games**  
**Take 2 Interactive**

# ACKNOWLEDGMENTS

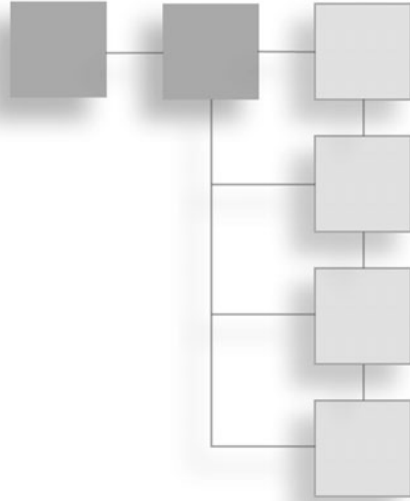


I am grateful to my wife, Jennifer, for giving me the time and space to write while also working full time, which takes away most of my free time. Thank you for being so supportive. I love you. It's hard to believe, but since the first edition of this book was published, we've added two more members to our family. Jeremiah and Kayleigh have welcomed Kaitlyn and Kourtney to our home in the past two years. I thank God for all of these blessings.

I am indebted to the hard working editors, artists, and layout specialists at Thomson Course Technology PTR and to all of the freelancers for doing such a fine job. Many thanks especially to Jenny Davidson, Brandon Penticuff, Mitzi Koontz, and Emi Smith. Thanks go to Joshua Smith for his technical review, which was invaluable. I believe you will find this a true gem of a game programming book due to all of their efforts.

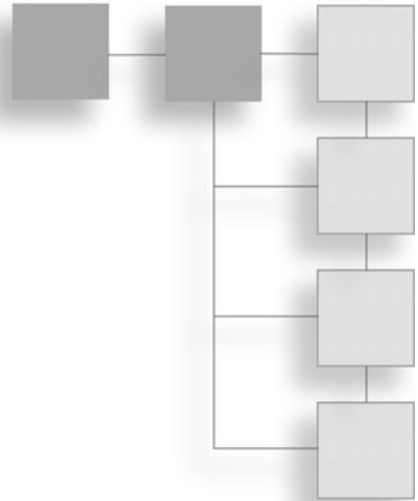


## ABOUT THE AUTHOR



**Jonathan S. Harbour** is a senior instructor of game development at the University of Advancing Technology ([www.uat.edu](http://www.uat.edu)) in Tempe, Arizona, where he teaches a variety of game programming courses. When not teaching others about games, writing about games, or playing games, he enjoys audio/video editing, wrenching on old Fords (and going to local car shows), and watching movies. His favorite game development tools are DarkBASIC, Allegro, and DirectX. Jonathan is the author of these recent books: *Game Programming All in One, Third Edition*; *DarkBASIC Pro Game Programming, Second Edition* (with Joshua Smith); *Beginning Java 5 Game Programming*; and *The Gadget Geek's Guide to Your Xbox 360*. Jonathan founded a small, independent game studio, Primeval Games, as a creative outlet for producing humorous casual games, and is working on several unique, new games, including a space shooter. He lives in Arizona with his wife, Jennifer, and four children: Jeremiah, Kayleigh, Kaitlyn, and newcomer Kourtney. He can be reached at [www.jharbour.com](http://www.jharbour.com).

# CONTENTS



	Introduction . . . . .	xv
<b>PART I</b>	<b>WINDOWS PROGRAMMING. . . . .</b>	<b>1</b>
<b>Chapter 1</b>	<b>Getting Started with Windows and DirectX . . . . .</b>	<b>3</b>
	Welcome to the Adventure! . . . . .	4
	Let's Talk About Compilers . . . . .	5
	What's Your Skill Level? . . . . .	7
	An Overview of Windows Programming . . . . .	10
	"Getting" Windows . . . . .	11
	Understanding Windows Messaging . . . . .	12
	Multi-Tasking . . . . .	13
	Multi-Threading . . . . .	16
	Event Handling . . . . .	17
	A Quick Overview of DirectX. . . . .	18
	What Is Direct3D? . . . . .	20
	What You Have Learned . . . . .	21
	Review Questions . . . . .	22
<b>Chapter 2</b>	<b>Windows Programming Basics . . . . .</b>	<b>25</b>
	The Basics of a Windows Program . . . . .	26
	Creating a Win32 Project . . . . .	26
	Understanding WinMain . . . . .	31
	The Complete WinMain . . . . .	33

	What You Have Learned . . . . .	36
	Review Questions . . . . .	37
	On Your Own . . . . .	37
<b>Chapter 3</b>	<b>Windows Messaging and Event Handling . . . . .</b>	<b>39</b>
	Writing a Full-Blown Windows Program . . . . .	40
	Understanding InitInstance . . . . .	44
	Understanding MyRegisterClass . . . . .	47
	Understanding WinProc . . . . .	50
	What You Have Learned . . . . .	55
	Review Questions . . . . .	56
	On Your Own . . . . .	57
<b>Chapter 4</b>	<b>The Real-Time Game Loop . . . . .</b>	<b>59</b>
	What Is a Game Loop? . . . . .	60
	The Old WinMain . . . . .	60
	WinMain and Looping . . . . .	62
	The GameLoop Project . . . . .	67
	Source Code for the GameLoop Program . . . . .	67
	What You Have Learned . . . . .	74
	Review Questions . . . . .	75
	On Your Own . . . . .	76
<b>PART II</b>	<b>DIRECTX PROGRAMMING . . . . .</b>	<b>77</b>
<b>Chapter 5</b>	<b>Your First DirectX Graphics Program . . . . .</b>	<b>79</b>
	Getting Started with Direct3D . . . . .	80
	The Direct3D Interfaces . . . . .	81
	Creating the Direct3D Object . . . . .	81
	Taking Direct3D for a Spin . . . . .	84
	Direct3D in Fullscreen Mode . . . . .	93
	What You Have Learned . . . . .	95
	Review Questions . . . . .	96
	On Your Own . . . . .	96
<b>Chapter 6</b>	<b>Bitmaps and Surfaces . . . . .</b>	<b>99</b>
	Surfaces and Bitmaps . . . . .	100
	The Primary Surfaces . . . . .	102
	Secondary Offscreen Surfaces . . . . .	102
	The Create_Surface Example . . . . .	105
	Loading Bitmaps from Disk . . . . .	112
	The Load_Bitmap Program . . . . .	113

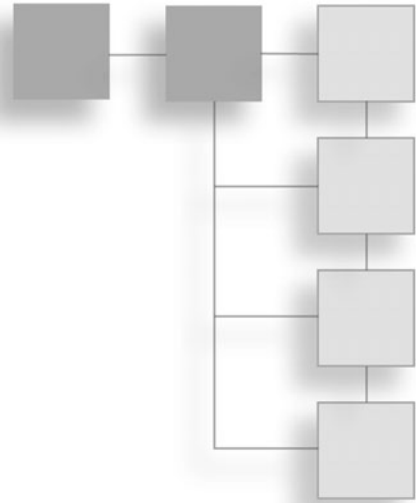
	What You Have Learned . . . . .	117
	Review Questions . . . . .	118
	On Your Own . . . . .	119
<b>Chapter 7</b>	<b>Drawing Animated Sprites . . . . .</b>	<b>121</b>
	Drawing Animated Sprites. . . . .	122
	The Anim_Sprite Project . . . . .	122
	Concept Art . . . . .	141
	Animated Sprites Explained . . . . .	142
	What You Have Learned . . . . .	147
	Review Questions . . . . .	148
	On Your Own . . . . .	149
<b>Chapter 8</b>	<b>Advanced Sprite Programming . . . . .</b>	<b>151</b>
	Drawing Transparent Sprites . . . . .	152
	Creating a Sprite Handler Object . . . . .	152
	Loading the Sprite Image . . . . .	154
	Drawing Transparent Sprites. . . . .	157
	Drawing an Animated Sprite . . . . .	164
	Working with Sprite Sheets . . . . .	165
	The Tiled_Sprite Program . . . . .	166
	Collision Detection . . . . .	170
	Testing for Collisions . . . . .	170
	The CollisionTest Program . . . . .	171
	What You Have Learned . . . . .	178
	Review Questions . . . . .	179
	On Your Own . . . . .	180
<b>Chapter 9</b>	<b>Jamming with DirectX Audio . . . . .</b>	<b>181</b>
	Using DirectSound . . . . .	182
	Initializing DirectSound . . . . .	183
	Creating a Sound Buffer. . . . .	184
	Loading a Wave File. . . . .	184
	Playing a Sound . . . . .	185
	Testing DirectSound . . . . .	186
	Creating the Project . . . . .	187
	Creating the DirectX Audio Support Files . . . . .	191
	Tweaking the Framework Code . . . . .	194
	Adding the Game Files. . . . .	195
	Running the Program. . . . .	201

	What You Have Learned . . . . .	202
	Review Questions . . . . .	203
	On Your Own . . . . .	203
<b>Chapter 10</b>	<b>Handling Input Devices . . . . .</b>	<b>205</b>
	The Keyboard . . . . .	206
	DirectInput Object and Device . . . . .	206
	Initializing the Keyboard . . . . .	208
	Reading Key Presses . . . . .	209
	The Mouse . . . . .	210
	Initializing the Mouse . . . . .	210
	Reading the Mouse . . . . .	212
	Paddle Game . . . . .	213
	The New Framework Code for DirectInput . . . . .	213
	The Paddle Game Source Code . . . . .	219
	Paddle Game Explained . . . . .	227
	What You Have Learned . . . . .	228
	Review Questions . . . . .	229
	On Your Own . . . . .	230
<b>Chapter 11</b>	<b>Tile-Based Scrolling Backgrounds . . . . .</b>	<b>231</b>
	Introduction to Scrolling . . . . .	232
	Introduction to Tile-Based Backgrounds . . . . .	233
	Backgrounds and Scenery . . . . .	233
	Creating Backgrounds from Tiles . . . . .	234
	Tile-Based Scrolling . . . . .	234
	Dynamically Rendered Tiles . . . . .	243
	The Tile Map . . . . .	244
	Creating a Tile Map Using Mappy . . . . .	245
	The DynamicScroll Project . . . . .	251
	What You Have Learned . . . . .	260
	Review Questions . . . . .	260
	On Your Own . . . . .	261
<b>PART III</b>	<b>3D PROGRAMMING . . . . .</b>	<b>263</b>
<b>Chapter 12</b>	<b>3D Graphics Fundamentals . . . . .</b>	<b>265</b>
	Introduction to 3D Programming . . . . .	266
	The Three Steps to 3D Programming . . . . .	267
	The 3D Scene . . . . .	267
	Moving to the Third Dimension . . . . .	272

	Grabbing Hold of the 3D Pipeline. . . . .	273
	The Vertex Buffer. . . . .	275
	Rendering the Vertex Buffer. . . . .	278
	Creating a Quad . . . . .	279
	The Textured Cube Demo . . . . .	282
	Modifying the Framework . . . . .	282
	The Cube_Demo Program. . . . .	288
	What's Next? . . . . .	294
	What You Have Learned . . . . .	295
	Review Questions . . . . .	296
	On Your Own. . . . .	296
<b>Chapter 13</b>	<b>Creating Your Own 3D Models with Anim8or. . . . .</b>	<b>299</b>
	Introducing Anim8or. . . . .	300
	Getting into 3D Modeling . . . . .	300
	Features. . . . .	302
	The Interface . . . . .	303
	Installing Anim8or . . . . .	303
	Using Anim8or . . . . .	304
	Stock Primitives . . . . .	305
	Manipulating Objects. . . . .	310
	Manipulating the Entire Scene . . . . .	314
	Creating the Car Model. . . . .	318
	The Wheels . . . . .	319
	The Frame . . . . .	330
	The Windows. . . . .	333
	The Headlights and Taillights . . . . .	334
	Creating a Scene. . . . .	338
	What You Have Learned . . . . .	340
	Review Questions . . . . .	341
	On Your Own. . . . .	342
<b>Chapter 14</b>	<b>Working with 3D Model Files. . . . .</b>	<b>343</b>
	Converting 3D Files. . . . .	344
	Converting 3DS to .X . . . . .	344
	Loading and Rendering a Model File. . . . .	351
	Loading an .X File . . . . .	352
	Rendering a Complete Model. . . . .	354
	The Load_Mesh Program . . . . .	355
	What's Next? . . . . .	360

What You Have Learned . . . . .	362
Review Questions . . . . .	362
On Your Own . . . . .	363
<b>Chapter 15 Complete 3D Game . . . . .</b>	<b>365</b>
Bash . . . . .	366
Playing the Game. . . . .	368
Creating the Models. . . . .	373
Printing Text Using a Bitmapped Font. . . . .	376
Simple 3D Collision Detection. . . . .	379
Bash Source Code. . . . .	380
What's Next? . . . . .	380
What You Have Learned . . . . .	381
Review Questions . . . . .	382
On Your Own . . . . .	382
<b>Index . . . . .</b>	<b>385</b>

# INTRODUCTION



This book will teach you the fundamentals of how to write games in the C++ language, using the powerful but intimidating DirectX 9 SDK. Game programming is a challenging subject that is not just difficult to master; it is difficult just to get started. This book takes away the mystery of game programming using the tools of the trade: C++ and DirectX. You will learn how to harness the power of Windows and DirectX to write both 2D and 3D games, with an especially strong emphasis on some of the more advanced topics in 3D programming for a beginning book.

You will learn how to write a simple Windows program. From there, you will learn about the key DirectX components: Direct3D, DirectSound, and DirectInput. You will learn how to make use of these key DirectX components while writing simple code that is easy to understand, at a pace that will not leave you behind. Along the way, you will put all of the new information gleaned from each chapter into a framework, or game library, that will be readily available to you in future chapters (as well as your own future game projects). After you have learned all that you need to know to write a simple game, you will do just that. And it is not just the usual sprite-based game either; it's a complete, fully functional 3D game, using collision detection, with real 3D models. A complete chapter will teach you just how to create your own models using the popular and free Anim8or modeling program (included on the CD-ROM).



## Where to Begin?

My philosophy for game development is neither limited nor out of reach for the average programmer. I want to really get down to business early on and not have to explain every function call in the standard C++ library. So you will want to begin learning C++ right now if you are not familiar with the language. There are certainly a lot of great products you can use that are as powerful (or more so) as the language used in this book. There are products like Blitz Basic (see *Game Programming for Teens* by Maneesh Sethi) and DarkBASIC (see *DarkBASIC Pro Game Programming, 2nd Edition* by Jonathan Harbour and Joshua Smith). These are two examples of game development tools that provide you with a complete package: compiler, editor, game library/engine, and the ability to produce a standalone Windows/DirectX game without the need for a runtime library of any kind. If you are fairly new to the C++ language or have no experience with it at all, I strongly suggest that you read a C primer first (such as *C Programming for the Absolute Beginner* by Michael Vine). I often use the terms “C” and “C++” interchangeably to avoid confusion, but most of the code in this book is actually just basic C rather than C++.

Why am I recommending so many books? Well, the books on BASIC are just mentioned in passing (as a subject that you may wish to pursue), while I do recommend that you read a C primer before continuing with this book. Game programming as a subject is not something that you just pick up after reading a single book. Although this book has everything you need to write simple 2D and 3D games (and granted it does cover a lot of useful information in that regard), no single volume can claim to cover everything because game development is a complex subject. I am confident that you will manage to follow along and grasp the concepts in this book just fine without one, but a C primer will give you a very good advantage before getting into Windows and DirectX programming. This book spends no time at all discussing the C language; it jumps right into Windows and DirectX code fairly quickly, followed by a new subject in each chapter!

This book was written in a progressive style that is meant to challenge you at every step, and relies on repetition rather than memorization. I don't cover a difficult subject just once and expect you to know it from that point on. Instead, I just present similar code sections in each program so you'll get the hang of it over time. The learning curve here is modeled after driving a car: once you have learned to use the accelerator and brake pedals, the actual process of learning to drive comes from practice. You wouldn't dare attempt to compete in a NASCAR race

after simply reading a driving book, would you? Of course not! But after many hours behind the wheel, you would at least be qualified to drive around the track.

I would rather you learn to draw a Bresenham line on your own than to copy someone else's texture-wrapped polygon code. There are a lot of things we will have to just take for granted in this book, because the goal is to teach the basics and prepare you for further study. But at the same time, I don't want to give you the impression that you can get by just by copying and pasting code to accomplish what you need for a particular game. On the contrary, the up-front learning curve is a challenge, and can be frustrating at times, but you have to get started somewhere, so my goal is to help you develop a love of learning and foster that love for video games that prompted you to pick up this book.

So, where to begin? If this book is going to teach you the basics of DirectX, so that you can write your own games, then we need to start with the basics of a Windows program.

## **What Will You Learn in This Book?**

This book will teach you how to write a Windows program, and from there, the sky's the limit! You will learn about DirectX; you will dive into Direct3D head-first and learn all about surfaces, textures, meshes, 3D models, and that is just the beginning!

You will learn how to interface with your computer's hardware using DirectX components, and use those hardware devices in your games!

Since this book is dedicated to teaching the basics of game programming, it will cover a lot of subjects very quickly, so you'll need to be on your toes! I use a casual writing style to make the subjects easy to understand and use repetition rather than memorization to nail the points home. You will learn by doing and you will not struggle with any one subject, because you will practice each topic several times throughout the book. Each chapter builds on the one before, but may be considered independent, so if there is any one subject that you are very interested in at the start, then feel free to skip around. However, the game framework built in this book does refer back to previous chapters, so I recommend reading it one chapter at a time.

This book spends a lot of time on 3D programming, but in order to get to the 3D material, there is a lot of information that must be covered first. Those topics are covered quickly so you will be learning some of the advanced topics in 3D

programming in no time. In order to load a 3D model, for instance, you will need to learn how to *create* a 3D model first, right? Well, you will learn just how to do that in this book!

Anim8or is a powerful 3D modeling program that is free and included on the CD-ROM that accompanies this book. You will learn how to use Anim8or in Chapter 13 to create a complete model of a car.

After you have learned the ropes of 3D modeling, you will also need to learn how to convert your 3D models to a format that Direct3D will understand. Chapter 14 explains how to convert the models exported from Anim8or to the Direct3D format.

## What Compiler Should You Use?

This book uses the C++ language and all examples are compiled with Microsoft Visual C++ 2003. You should be able to compile and run the programs using another Windows compiler such as Borland C++Builder or with another version of Visual C++ (6.0 and later should work fine). You may also use the free Visual C++ 2005 Express Edition, available for download from Microsoft's Web site.

## What About the Programming Language?

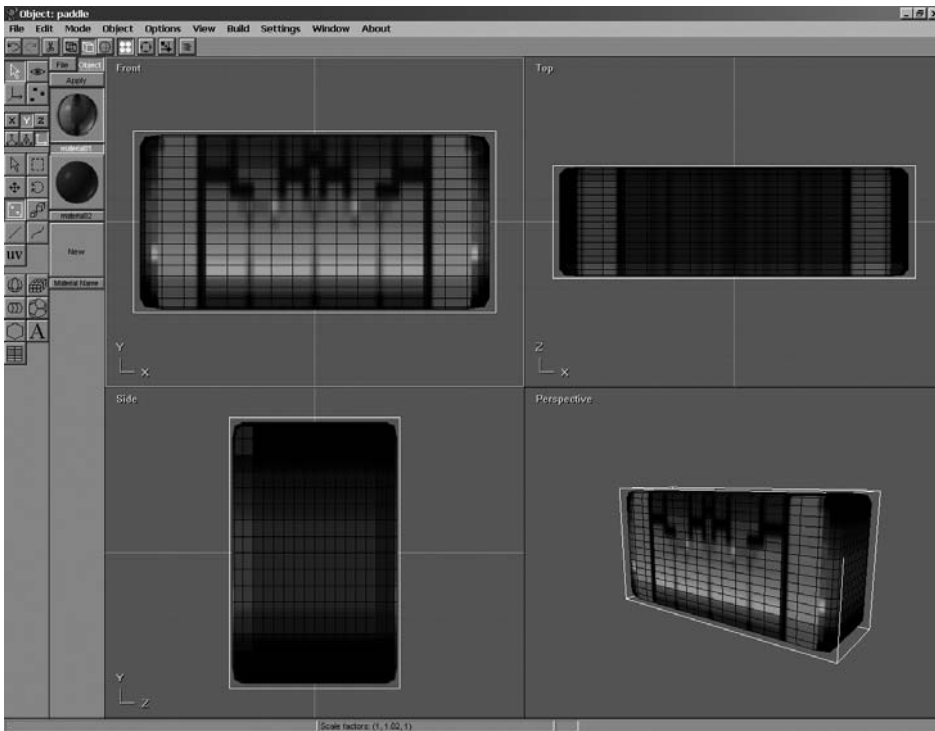
This book focuses on the C++ language. This book is not a primer on the C++ language, but rather makes use of this very powerful, low-level language to write games. The examples and source code are mostly C, except for the use of some specific C++ here and there. You will get by just fine with a basic understanding of the C language. Just know that I do not teach the language in this book—we get down to business writing games very quickly and do not have time for a tutorial on C/C++ programming.

As such, you *do* need to know C in advance (preferably, C++). If this is your first experience with the C language, and you have not used it before, I'll be honest with you, you will have a very hard time with the source code in this book. If you feel that you are up to the challenge, then you *might* be able to wade through the C code and make some sense out of it. But I want to warn you in advance: I don't spend even a single paragraph trying to teach you anything about the C language! This book is about game programming, and it assumes that you already know C. I recommend that you acquire a C primer to read before delving into this book, or to keep handy for those parts that may confuse you.

## What About a Complete Game?

*Beginning Game Programming, Second Edition* is not a tutorial on how to program in C, and not a DirectX reference. This book is all about game programming. You will learn the skills to write a complete 3D game in C and DirectX 9 called *Bash*. *Bash* demonstrates wireframe and solid rendering with materials and textures using Direct3D, and uses real 3D models created with Anim8or.

Creating this game is not just a matter of typing in some source code and compiling it, then away you go. On the contrary, you need to create your own 3D models for this game. I encourage this throughout the book, because if you want to master game programming, you need to become proficient with a modeling package like Anim8or (which is almost as feature rich as 3ds max and Maya, for our purposes here). You will actually see how the artwork for *Bash* is created. Since you learn how to create your own models in Chapter 13, you will be able to enhance and modify *Bash* to suit your own tastes by modifying the 3D models in Anim8or. How would you like to add your own photos to be used as textures in the game? No problem, you will learn how to do things like that in this book.



You will learn how the models for *Bash* were created.