

XNA 2.0 Game Programming Recipes

A Problem-Solution Approach



Riemer Grootjans

XNA 2.0 Game Programming Recipes: A Problem-Solution Approach

Copyright © 2008 by Riemer Grootjans

All rights reserved. No part of this work 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 the prior written permission of the copyright owner and the publisher.

ISBN-13 (pbk): 978-1-59059-925-9

ISBN-10 (pbk): 1-59059-925-X

ISBN-13 (electronic): 978-1-4302-0514-2

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Trademarked names may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, we use the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Lead Editor: Ewan Buckingham

Technical Reviewer: Fabio Claudio Ferracchiati

Editorial Board: Clay Andres, Steve Anglin, Ewan Buckingham, Tony Campbell, Gary Cornell,

Jonathan Gennick, Matthew Moodie, Joseph Ottinger, Jeffrey Pepper, Frank Pohlmann,

Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh

Project Manager: Beth Christmas

Copy Editor: Kim Wimpsett

Associate Production Director: Kari Brooks-Copony

Production Editor: Ellie Fountain

Compositor: Susan Glinert

Proofreader: April Eddy

Indexer: Broccoli Information Management

Artist: April Milne

Cover Designer: Kurt Krames

Manufacturing Director: Tom Debolski

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders-ny@springer-sbm.com, or visit <http://www.springeronline.com>.

For information on translations, please contact Apress directly at 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705. Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or visit <http://www.apress.com>.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales—eBook Licensing web page at <http://www.apress.com/info/bulksales>.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

The source code for this book is available to readers at <http://www.apress.com>.

*To Elisa, the most lovely woman I know. Without your love and support,
this would not have been possible.*

To my parents, for giving me each and every opportunity.

To my friends and co-workers, for their support and laughs.

Contents at a Glance

About the Authorxi
About the Technical Reviewerxiii
Acknowledgments	xv
Introduction	xvii
■ CHAPTER 1 Getting Started with XNA 2.0	1
■ CHAPTER 2 Setting Up Different Camera Modes in Your 3D World	25
■ CHAPTER 3 Working with 2D Images/Textures in XNA 2.0	145
■ CHAPTER 4 Working with Models	247
■ CHAPTER 5 Getting the Most Out of Vertices	355
■ CHAPTER 6 Adding Light to Your Scene in XNA 2.0	499
■ CHAPTER 7 Adding Sounds to Your XNA 2.0 Project	559
■ CHAPTER 8 Networking in XNA 2.0	571
■ INDEX	599

Contents

About the Author	xi
About the Technical Reviewer	xiii
Acknowledgments	xv
Introduction	xvii

■ CHAPTER 1 **Getting Started with XNA 2.0** 1

1-1. Install XNA Game Studio 2.0	1
1-2. Start Your First XNA 2.0 Project	3
1-3. Deploy Your XNA2.0 Game on Xbox 360	6
1-4. Deploy Your XNA2.0 Game on Another PC	8
1-5. Customize Game Loop Timing	10
1-6. Make Your Code Plug-and-Play Using GameComponents	11
1-7. Allow Your GameComponents to Communicate with Each Other by Implementing GameServices	15
1-8. Save and Load Data to/from a File	19

■ CHAPTER 2 **Setting Up Different Camera Modes in Your 3D World** ... 25

2-1. Set Up the Camera: Position, Target, and View Frustum	25
2-2. Specify the Target of Your Camera	33
2-3. Create a First-Person Shooter Camera: A Quake-Style Camera ...	39
2-4. Create a Freelancer-Style Camera: Full 3D Rotation Using Quaternions	45
2-5. Check Whether an Object Is in Sight of the Camera	52
2-6. Detect Camera Collision Against Models, Walls, or Terrains	57
2-7. Create a Camera Fly-by Action	60
2-8. Remove the Solid Background Color: Skybox	70
2-9. Draw Only the Parts of the Scene That Are in Sight of the Camera: Octree	82
2-10. Use a Quadtree to Hide Parts of a Grid That Are Not in Sight ...	104
2-11. Set Up a Post-Processing Framework	113
2-12. Create a Blur/Glow Post-Processing Effect	125
2-13. Define an Orthogonal Projection Matrix to Generate a Depth Map of the Scene	135

CHAPTER 3	Working with 2D Images/Textures in XNA 2.0	145
3-1.	Display 2D Images: Load and Render Images	
	Using the SpriteBatch Class	146
3-1.	Rotate, Scale, and Mirror an Image	151
3-3.	Render Transparent Images Using Layers	153
3-4.	Consider Performance When Using the SpriteBatch Class	157
3-5.	Display Text	160
3-6.	Create a 2D Menu Interface	162
3-7.	Create a Texture, Define the Color of Each Pixel,	
	Save a Texture to a File	176
3-8.	Render the Scene into a Texture	178
3-9.	Extend the Image Content Processor	182
3-10.	Extend the Image Content Processor: Grayscale Conversion	
	and Processor Parameters	195
3-11.	Make Your Scene More Impressive with Billboarding:	
	Render 2D Images in a 3D World So They Always Face	
	the Camera	199
3-12.	Create a 3D Explosion Effect/Simple Particle System	220
3-13.	Create a Mirror: Projective Texturing	233
CHAPTER 4	Working with Models	247
4-1.	Load and Render a Model Using the BasicEffect Class	248
4-2.	Set Different World Matrices for Different Objects,	
	Combining World Matrices	252
4-3.	Find the Rotation Angle Corresponding to a Direction	258
4-4.	Use Acceleration to Control Velocity	260
4-5.	Construct the Global BoundingSphere Around a Model	265
4-6.	Scale the Model to a Predefined Size	267
4-7.	Render a Model Using Custom Effects and	
	Custom Textures (Easy Approach)	269
4-8.	Visualize the Bone Structure of a Model	274
4-9.	Make the Bones Move Individually: Model Animation	281
4-10.	Use BoundingSpheres for Basic Model Collision Detection	286
4-11.	Use Ray-Traced Collision Detection for Small/Fast Objects	292
4-12.	Extend the Model Content Processor to	
	Load Custom Effects (Clean Approach)	295
4-13.	Gain Direct Access to Vertex Position Data by	
	Extending the Model Processor	302

4-14. Gain Direct Access to Vertex Position Data of Each ModelMesh by Extending the Model Processor	307
4-15. Gain Direct Access to Vertex Position Data by Defining a Custom TypeWriter and TypeReader	312
4-16. Store Multiple Objects in the Tag Property by Defining a Custom TypeWriter and TypeReader	322
4-17. Correctly Tilt a Model Corresponding to the Terrain Underneath	329
4-18. Detect Ray-Model Collisions Using Per-Triangle Checks	340
4-19. Detect Whether the Pointer Is Over a Model	349

■ CHAPTER 5 **Getting the Most Out of Vertices** 355

5-1. Render Triangles, Lines, and Points in a 3D World	356
5-2. Apply a Texture to Your Triangles	367
5-3. Remove Redundant Vertices Using Indices	375
5-4. Store Your Vertices and Indices in the Memory of Your Graphics Card Using a VertexBuffer and an IndexBuffer	379
5-5. Store Your Vertices in a DynamicVertexBuffer for Frequently Updated Data	384
5-6. Enable Backface Culling: What It Is and What It Can Do for You	387
5-7. Automatically Calculate the Normals for All Vertices in a VertexBuffer	391
5-8. Create a Terrain Based on a VertexBuffer and an IndexBuffer ...	397
5-9. Calculate the Exact Height of a Terrain Between Vertices Using Bilinear Interpolation	407
5-10. Calculate the Collision Point Between the Pointer and the Terrain: Surface Picking	413
5-11. Extend the TextureProcessor to Generate a Terrain Object from an Image: Advanced DOM Objects	420
5-12. Write a Custom Content Importer: Loading a Terrain from a CSV File	432
5-13. Load Data from an XML File	439
5-14. Create Your Own Vertex Format	445
5-15. Introducing Bump Mapping: Fixed Normal	455
5-16. Adding Per-Pixel Detail by Bump Mapping in Tangent Space ...	460
5-17. Add an Ocean to Your 3D World	472
5-18. Apply Catmull-Rom Interpolation in 3D to Generate Additional Vertices	485
5-19. Create the Vertices for a Racing Track	489

CHAPTER 6	Adding Light to Your Scene in XNA 2.0	499
6-1.	Define Normals and Use the BasicEffect	500
6-2.	Share Normals Between Vertices	506
6-3.	Add Higher Detail to Your Lighting: Per-Pixel Lighting	511
6-4.	Add Specular Highlights to Reflective Surfaces	514
6-5.	Add HLSL Vertex Shading	516
6-6.	Define a Point Light Using HLSL	522
6-7.	Add HLSL Per-Pixel Lighting	524
6-8.	Define a Spotlight Using HLSL	528
6-9.	Add HLSL Specular Highlights	530
6-10.	Add Multiple Lights to Your Scene Using Deferred Shading	535
6-11.	Add Shadowing Capability to Your Deferred Shading Engine	551
CHAPTER 7	Adding Sounds to Your XNA 2.0 Project	559
7-1.	Play Simple .wav Sound Files	559
7-2.	Loop Sounds	563
7-3.	Play Sounds from a 3D Location Relative to the Camera: 3D Sound	567
CHAPTER 8	Networking in XNA 2.0	571
8-1.	Sign In for Networking Services	571
8-2.	Create a Network Session	574
8-3.	Join a Networking Session	579
8-4.	Send/Receive Data Over the Network	584
8-5.	Search for Networking Sessions Asynchronously	592
8-6.	Move from the Lobby to the Actual Game	595
INDEX		599

About the Author



RIEMER GROOTJANS received a degree in electronic engineering with a specialization in informatics at the Vrije Universiteit Brussel in Brussels, Belgium. He is currently working as a member of a research team toward a Ph.D. degree. The goal of the team is to develop a real-time 3D depth-sensing camera, and he is responsible for (amongst other things) the visualization of the 3D data.

For a few years, Riemer has been maintaining a web site with tutorials for DirectX. Since the launch of XNA in December 2006, he has ported all his content to XNA and is helping more than 1,000 people on their path to XNA success every day. In July 2007, he received the Microsoft MVP Award for his contributions to the XNA community.

About the Technical Reviewer

■ **FABIO CLAUDIO FERRACCHIATI** is a senior consultant and a senior analyst/developer. He works for Brain Force (<http://www.brainforce.com>) in its Italian branch (<http://www.brainforce.it>). He is a Microsoft Certified Solution Developer for .NET, a Microsoft Certified Application Developer for .NET, and a Microsoft Certified Professional, and he is a prolific author and technical reviewer. Over the past ten years he has written articles for Italian and international magazines and coauthored more than ten books on a variety of computer topics. You can read his LINQ blog at <http://www.ferracchiati.com>.

Acknowledgments

I would like to express my appreciation and thankfulness to the skillful group of professionals at Apress that helped me complete this book.

Thanks to Fabio Claudio Ferracchiati for his efforts as technical editor of this book. He also guided me in making sure the whole XNA Framework was covered by the contents of this book. Thanks to Kim Wimpsett for carefully correcting my text and making it conform to the Apress style. Thanks to Ellie Fountain, who made the images and documents printer-ready. Last but not least, thanks to Beth Christmas for setting the deadlines and for resolving everyone's problems. You really made a difference!

Furthermore, I would like to thank Danc from <http://lostgarden.com> for the 2D artwork used in my book and code. There are some real gems to be found on your site!

And last but definitely not least, I thank XanTium from <http://x-scene.com> for providing me with the necessary hardware on which to test my code.

Introduction

When Microsoft released XNA in December 2006, it immediately became clear that this new technology would have a major impact on the possibilities for game developers. XNA was designed from the ground up with ease of use in mind, while not sacrificing performance or capabilities to achieve this goal. As a bonus, any game you create in XNA for the PC also runs on the Xbox 360 console!

In the span of one year, a large user community has grown around XNA. You can find code examples on a vast number of sites, ask your questions in one of the lively forums, or even meet local people who share the same passion in one of the XNA User Groups.

Whether you want to get up to speed with XNA quickly or you have tried some of the tutorial sites and are looking for the next step, this book is for you. With almost 100 recipes dealing with various challenges you may encounter during your journey with XNA, this book covers each corner of the XNA Framework.

The first recipes of the chapters in this book explain some stand-alone concepts and have been kept as clear as possible. As an example, a recipe explaining how to load a 3D Model from a file and render it to the screen will not render any trees in the background to make the final result look nicer, because this would clutter the code and make it more complex than it should be.

On the other hand, each chapter ends with some recipes that combine all you've learned thus far into something new and powerful. As such, you can step through the recipes, building up your experience in XNA as you move to the next recipe.

This book explains the functionality of the XNA 2.0 Framework. If any updates are made to the XNA Framework in the future, I will update the code for this book and make it available for download from my web site at <http://www.riemers.net/>.

If you have any questions regarding the text or code examples found in this book, I kindly invite you to post them on the forum on my site so you can get an answer as soon as possible.

The XNA Framework is roughly dividable into three parts. The main part contains your XNA project and its code. Next in line is the content pipeline, a flexible component allowing you to preprocess any art assets you want to load into your XNA project. Last, but definitely not least, are the HLSL effects, which are used mainly to improve the visual quality of the final image you render to the screen. Although each chapter starts with some recipes that cover the XNA functionality related to the chapter, this book can also be used as a detailed guide to the content pipeline and to HLSL effects, as explained in the following sections.

Content Pipeline

If you're interested in getting into the XNA content pipeline, I advise you to read the following recipes in this order:

3-9. Extend the Image Content Processor

3-10. Extend the Image Content Processor: Grayscale Conversion and Processor Parameters

- 4-13. Gain Direct Access to Vertex Position Data by Extending the Model Processor
- 4-14. Gain Direct Access to Vertex Position Data of Each ModelMesh by Extending the Model Processor
- 4-15. Gain Direct Access to Vertex Position Data by Defining a Custom TypeWriter and TypeReader
- 4-16. Store Multiple Objects in the Tag Property by Defining a Custom TypeWriter and TypeReader
- 5-13. Load Data from an XML File
- 5-12. Write a Custom Content Importer: Loading a Terrain from a CSV File
- 5-11. Extend the TextureProcessor to Generate a Terrain Object from an Image: Advanced DOM Objects

HLSL

This book also contains a lot of HLSL samples. You can follow these recipes in this order:

- 6-5. Add HLSL Vertex Shading
- 6-6. Define a Point Light Using HLSL
- 6-7. Add HLSL Per-Pixel Lighting
- 6-8. Define a Spotlight Using HLSL
- 6-9. Add HLSL Specular Highlights
- 6-10. Add Multiple Lights to Your Scene Using Deferred Shading
- 5-14. Create Your Own Vertex Format
- 5-15. Introducing Bump Mapping: Fixed Normal
- 5-16. Adding Per-Pixel Detail by Bump Mapping in Tangent Space
- 3-11. Make Your Scene More Impressive with Billboarding: Render 2D Images in a 3D World So They Always Face the Camera
- 3-12. Create a 3D Explosion Effect/Simple Particle System
- 2-13. Define an Orthogonal Projection Matrix to Generate a Depth Map of the Scene
- 3-13. Create a Mirror: Projective Texturing
- 6-11. Add Shadowing Capability to Your Deferred Shading Engine
- 5-17. Add an Ocean to Your 3D World