XNA 3.0 Game Programming Recipes: A Problem-Solution Approach

Copyright © 2009 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): 978-1-4302-1855-5

ISBN-13 (electronic): 978-1-4302-1856-2

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
Development Editor: Joohn Choe
Technical Reviewer: Fabio Claudio Ferracchiati
Editorial Board: Clay Andres, Steve Anglin, Mark Beckner, Ewan Buckingham, Tony Campbell, Gary Cornell, Jonathan Gennick, Michelle Lowman, Matthew Moodie, Jeffrey Pepper, Frank Pohlmann, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh
Project Manager: Richard Dal Porto
Copy Editors: Heather Lang, Kim Wimpsett
Associate Production Director: Kari Brooks-Copony
Production Editor: Elizabeth Berry
Compositor: Linda Weidemann, Wolf Creek Publishing Services
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`. You may need to answer questions pertaining to this book in order to successfully download the code.

Contents

About the Author. xxiii

About the Technical Reviewer xxv

Acknowledgments xxvii

Introduction xxix

CHAPTER 1 **Getting Started with XNA 3.0** 1

1-1. Install XNA Game Studio 3.0 1
1-2. Start Your First XNA 3.0 Project 2
1-3. Deploy Your XNA 3.0 Game on Xbox 360 5
1-4. Deploy Your XNA 3.0 Game on the Zune. 8
1-5. Deploy Your XNA 3.0 Game on Another PC 10
1-6. Customize Game Loop Timing 12
1-7. Make Your Code Plug-and-Play Using GameComponents. 14
1-8. Allow Your GameComponents to Communicate with
Each Other by Implementing GameServices 18
1-9. Save and Load Data to or from a File 23

CHAPTER 2 **Setting Up Different Camera Modes in Your 3D World** . . . 29

2-1. Set Up the Camera: Position, Target, and View Frustum 30
2-2. Specify the Target of Your Camera 37
2-3. Create a First-Person Shooter Camera: A Quake-Style Camera. . . . 44
2-4. Create a Freelancer-Style Camera: Full 3D Rotation
Using Quaternions 50
2-5. Check Whether an Object Is in Sight of the Camera. 57
2-6. Detect Camera Collision Against Models, Walls, or Terrains. 63
2-7. Create a Camera Fly-by Action. 66
2-8. Remove the Solid Background Color: Skybox 76
2-9. Draw Only the Parts of the Scene That Are in Sight
of the Camera: Octree 88
2-10. Use a Quadtree to Hide Parts of a Grid That Are Not in Sight 111
2-11. Create a Real-Time Camera-Dependant Optimally
Adapting Mesh (ROAM) Terrain 121

2-12. Set Up a Post-Processing Framework. 143
2-13. Create a Blur/Glow Post-Processing Effect 155
2-14. Write a Custom Content Importer 166

CHAPTER 3 **Working with 2D Images/Textures in XNA 3.0** 173

3-1. Display 2D Images: Load and Render Images Using the SpriteBatch Class 174
3-2. Rotate, Scale, and Mirror an Image. 180
3-3. Render Transparent Images Using Layers 183
3-4. Consider Performance When Using the SpriteBatch Class 187
3-5. Display Text 190
3-6. Create a 2D Menu Interface 192
3-7. Create a Texture, Define the Color of Each Pixel, Save a Texture to a File. 206
3-8. Render the Scene into a Texture 209
3-9. Extend the Image Content Processor. 213
3-10. Extend the Image Content Processor: Grayscale Conversion and Processor Parameters 226
3-11. Make Your Scene More Impressive with Billboarding: Render 2D Images in a 3D World So They Always Face the Camera 231
3-12. Create a 3D Explosion Effect/Simple Particle System 253
3-13. Create a Mirror: Projective Texturing 266

CHAPTER 4 **Working with Models** 281

4-1. Load and Render a Model Using the BasicEffect Class 282
4-2. Set Different World Matrices for Different Objects, Combining World Matrices 286
4-3. Find the Rotation Angle Corresponding to a Direction 293
4-4. Use Acceleration to Control Velocity 295
4-5. Construct the Global BoundingSphere Around a Model 299
4-6. Scale the Model to a Predefined Size 302
4-7. Render a Model Using Custom Effects and Custom Textures (Easy Approach) 304
4-8. Visualize the Bone Structure of a Model. 309
4-9. Make the Bones Move Individually: Model Animation 316
4-10. Use BoundingSpheres for Basic Model Collision Detection. 321
4-11. Use Ray-Traced Collision Detection for Small/Fast Objects 327
The Problem 327

4-12. Extend the Model Content Processor to Load Custom Effects (Clean Approach) 331
4-13. Gain Direct Access to Vertex Position Data by Extending the Model Processor 339
4-14. Gain Direct Access to Vertex Position Data of Each ModelMesh by Extending the Model Processor 344
4-15. Gain Direct Access to Vertex Position Data by Defining a Custom TypeWriter and TypeReader 349
4-16. Store Multiple Objects in the Tag Property by Defining a Custom TypeWriter and TypeReader 360
4-17. Correctly Tilt a Model According to the Terrain Underneath 366
4-18. Detect Ray-Model Collisions Using Per-Triangle Checks 378
4-19. Detect Whether the Pointer Is Over a Model 388

CHAPTER 5 **Getting the Most Out of Vertices** 393

5-1. Render Triangles, Lines, and Points in a 3D World 394
5-2. Apply a Texture to Your Triangles 405
5-3. Remove Redundant Vertices Using Indices 413
5-4. Store Your Vertices and Indices in the Memory of Your Graphics Card Using a VertexBuffer and an IndexBuffer 417
5-5. Store Your Frequently Updated Vertices in a DynamicVertexBuffer 423
5-6. Enable Backface Culling: What It Is and What It Can Do for You 425
5-7. Automatically Calculate the Normals for All Vertices in a VertexBuffer 429
5-8. Create a Terrain Based on a VertexBuffer and an IndexBuffer 436
5-9. Calculate the Exact Height of a Terrain Between Vertices Using Bilinear Interpolation 446
5-10. Calculate the Collision Point Between the Pointer and the Terrain: Surface Picking 452
5-11. Load Data from an XML File 460
5-12. Create Your Own Vertex Format 466
5-13. Work with Bump Mapping: Fixed Normal 476
5-14. Add Per-Pixel Detail by Bump Mapping in Tangent Space 481
5-15. Add an Ocean to Your 3D World 493
5-16. Apply Catmull-Rom Interpolation in 3D to Generate Additional Vertices 507
5-17. Create the Vertices for a Racing Track 511
The Problem 511

CHAPTER 6 **Adding Light to Your Scene in XNA 3.0** 521

6-1. Define Normals and Use the BasicEffect. 522
6-2. Share Normals Between Vertices. 528
6-3. Add Higher Detail to Your Lighting: Per-Pixel Lighting 533
6-4. Add Specular Highlights to Reflective Surfaces. 536
6-5. Add HLSL Vertex Shading 538
6-6. Define a Point Light Using HLSL. 545
6-7. Add HLSL Per-Pixel Lighting 547
6-8. Define a Spotlight Using HLSL 551
6-9. Add HLSL Specular Highlights 553
6-10. Add Multiple Lights to Your Scene Using Deferred Shading 558
6-11. Add Shadowing Capability to Your Deferred Shading Engine 574

CHAPTER 7 **Adding Sounds to Your XNA 3.0 Project** 583

7-1. Play and Control Simple .wav Sound Files 583
7-2. Play MP3 or WMA Sound Files 585
7-3. Play Simple .wav Sound Files Through XAct 586
7-4. Loop Sounds 590
7-5. Play Sounds from a 3D Location Relative to the Camera:
3D Sound 594

CHAPTER 8 **Networking in XNA 3.0** 599

8-1. Sign In for Networking Services 599
8-2. Create a Network Session 602
8-3. Join a Network Session 608
8-4. Send/Receive Data Over the Network 614
8-5. Search for Networking Sessions Asynchronously 622
8-6. Add Rich Presence Information 626
8-7. Move from the Lobby to the Actual Game 627

INDEX 631