Building XNA 2.0 Games

A Practical Guide for Independent Game Development

James Silva and John Sedlak

Apress®

Building XNA 2.0 Games: A Practical Guide for Independent Game Development

ISBN-13 (pbk): 978-1-4302-0979-9

ISBN-13 (electronic): 978-1-4302-0980-5

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

Lead Editor: Ewan Buckingham
Technical Reviewer: Fabio Claudio Ferracchiati
Editorial Board: Clay Andres, Steve Anglin, Ewan Buckingham, Tony Campbell, Gary Cornell, Jonathan Gennick, Kevin Goff, Matthew Moodie, Joseph Ottinger, Jeffrey Pepper, Frank Pohlmann, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh
Project Manager: Beth Christmas
Copy Editor: Marilyn Smith
Associate Production Director: Kari Brooks-Copony
Production Editor: Ellie Fountain
Compositors: Susan Glinert and Octal Publishing, Inc.
Proofreader: Nancy Sixsmith
Indexer: Carol Burbo
Artist: Kinetic Publishing Services, LLC
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

This book is dedicated to my mom and dad,
who were always supportive of my game development obsession.
—James Silva

Contents at a Glance

Contents

About the Authors

JAMES SILVA has been creating games as a hobbyist developer for nearly a decade, but he never took himself quite seriously enough until his latest work, *The Dishwasher: Dead Samurai*, got some attention. The Dishwasher won the Microsoft Dream-Build-Play 2007 contest and earned James an Xbox Live Arcade contract. He was approached with the concept of creating a book focused on techniques used to create The Dishwasher.

James holds a Master's Degree in Computer Science from State University of New York Institute of Technology. He lives in Utica, New York, with two cats who he swears are trying to kill him. James is still hard at work on The Dishwasher, which will soon be making its debut on Xbox Live Arcade.

JOHN SEDLAK, a Microsoft MVP for XNA/DirectX, got his start in game development when he was just 11 years' old, with the help of Microsoft's Visual Basic. After completing a few games with BitBlting techniques, it was time to move on and learn the .NET Framework and all DirectX had to offer. Since then, John has placed a great deal of effort into understanding the design of frameworks and engines. From the first release of the XNA Framework, he has worked to grow the community through tutorials, code snippets, and complete open source games, such as GW3 and Domination.

In his spare time, John enjoys cycling on the open road and driving long distances, and has even been known to take a few photos along the way.

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.

Fabio 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 acknowledge John Sedlak, who saved this book from certain doom, as well as all of the great guys in the XNA community and Microsoft XNA team, who helped me with all of my stupid programming questions. (That is actually the term used—“stupid programming question”—and it is a question that one should not have to ask if one has been approached to write a book about the subject.)

James Silva

There is an incredibly long list of people who should be thanked—a list that would probably be longer than this book.

First and foremost, I would like to thank James for developing The Dishwasher, an amazing game that truly deserves all the honors it has received. I look forward to losing many nights’ sleep playing the game on my Xbox. I would also like to give thanks to the people behind the scenes at Apress. They truly are an amazing team of people, who have been incredibly patient while we strived for excellence.

Special thanks to all the hard-working developers and readers out there. Without you, this book could not exist. I hope you all learn something from this book, and I hope many more take what we cover and produce some original and amazing games with XNA.

John Sedlak

Introduction

We're in an amazing era of video games; high-definition, complex shader-powered, highly immersive 3D content is the norm. The games industry is bombarded by titles of incredible quality month after month. While the end product is great for gamers, it can be a bit disheartening to aspiring game developers with great ambitions and little experience.

Being in this crazy era, it's easy to make a number of mistakes while trying to jump into game development. Most are due to not really fully grasping the scope of a game development undertaking. For instance, it's easy to look at a lot of big-name games and start thinking in terms of cut scenes; or, a bit worse, to start thinking of massive multiplayer anything. Creating something simple, like a bouncing sprite, and then getting overwhelmed while trying to introduce bigger game-play concepts is a fairly common pitfall. James will readily admit to making all of the main mistakes at one point or another (though to be fair, it was in an era before MMORPGs).

When we set out to make this book, we intended to describe the process of creating a game very much like James's game, The Dishwasher: Dead Samurai—a platforming, combat-heavy 2D game with good controls, clean animation, and polished presentation. We could have introduced you to a smattering of math-intensive 3D concepts like BSP trees and volumetric lighting, but we wanted to give you something you can easily be productive with, because that's the fun part of game development. And that's the essence of what we're doing here: having fun. That's why we got into this business in the first place.

In this book, we take all of the main aspects of development from The Dishwasher and put them into a new game we'll be making called Zombie Smashers XNA. We'll take little, chapter-sized modules of functionality—things like map and character editors, basic platforming, particle effects, exploding zombie heads, and so on—and really give you a feel for what we're doing and, more important, what you can do. When it's all said and done, you'll have an excellent foundation for going anywhere with any sort of game of this scope: puzzle platformer, coin-op style beat-'em-up, story-driven role-playing game, and so on. Just don't expect to learn how to make a first-person shooter (FPS) here. Of course, that's not to say that the fundamentals we'll cover in this book won't help you should you decide to confront something as ambitious as an FPS (still, there's a reason most well-funded FPS developers don't use in-house engines!).

We'll be using Microsoft XNA Game Studio 2.0 to build a side-scrolling beat-em-'up game. XNA 2.0 is a great framework for game programming. It is extremely powerful, yet well suited for amateur, independent, and hobbyist developers. This book, of course, is written by amateur/indie/hobbyist developers for amateur/indie/hobbyist developers. Throughout the next several hundred pages, you'll get to see XNA really shine in this respect. We'll be focusing on techniques for good presentation and fast development, such as through fluid animation and eye-catching particle systems, where you'll see the most payoff for time invested.

We'll start off by covering some programming basics, and then jump right in to XNA with our version of a Hello World program: XNAPong! After the brief, two-chapter crash course on all things basic, we'll kick off the start of our Zombie Smashers XNA game with a map editor and character editor, and then start working directly with our game. We'll implement a solid platforming engine, particle systems, audio, and menus, before moving on to some advanced stuff like postprocessing effects and networking.

The really nice bit is that you can download the final projects now. In fact, you had better do it right away. The link is `http://apress.com/book/view/1430209798`.

This way, you'll be able to see exactly where we're headed before we get there. We find it kind of annoying and troublesome to keep writing code without getting much visual payoff. We like to see what we're doing! So snag the code online, fire up Zombie Smashers XNA in Visual Studio, run it on Windows, and see where we're headed. With all of the fully completed projects in hand, you shouldn't have to feel in the dark when we throw hundreds of lines of convoluted tools, particles, and who knows what else at you in the chapters to come.

Of course, we will skip around a lot—more in some chapters than in others. That's just the nature of the beast. We may want to add a bit of functionality to one area, but in doing so, we find we need to update a tool, introduce some global states, and so on. So bear this in mind while following the final projects: there may be code that the text doesn't cover yet. It's safe to ignore; we'll get to it all eventually.

All that said, it's probably safe to dive in!