Microsoft

3D PROGRAMMING FOR WINDOWS®

WINDOWS 3D 编程

THREE-DIMENSIONAL GRAPHICS PROGRAMMING FOR THE WINDOWS PRESENTATION FOUNDATION

China Edition
Published by Shanghai World Publishing Corporation

世界图书出版公司

Charles Petzold

Microsoft®

3D Programming for Windows®:

Three-Dimensional Graphics Programming for the Windows Presentation Foundation

Charles Petzold

图书在版编目(CIP)数据

Windows 3D 编程:英文/(美)佩特索德著. - 上海:上海世界图书出版公司,2009. 1

ISBN 978 - 7 - 5062 - 9170 - 5

Ⅰ. W… Ⅱ. 佩… Ⅲ. 三维 - 动画 - 图形软件,Windows3D - 教材 - 英文 Ⅳ. TP391. 41

中国版本图书馆 CIP 数据核字(2008)第 180649 号

Windows 3D 编程

[美]查尔斯·佩特索德 著

上海世界图书出版公司 出版发行
上海市尚文路 185 号 B 楼
邮政编码 200010
(公司电话:021 - 63783016 转发行部)
上海竟成印务有限公司印刷
如发现印装质量问题,请与印刷厂联系
(质检科电话:021 - 56422678)
各地新华书店经销

开本:787 × 960 1/16 印张:28 字数:900 000
2009 年 1 月第 1 版 2009 年 1 月第 1 次印刷
ISBN 978 - 7 - 5062 - 9170 - 5/T · 185
图字:09 - 2008 - 627 号
定价:188. 00 元
http://www. wpcsh. com. cn
http://www. mspress. com. cn

Introduction

Microsoft Windows Vista is the first version of Windows to have built-in support for three-dimensional graphics. This 3D graphics support is integrated with the Microsoft Windows Presentation Foundation (WPF), the client application programming interface (API) that was introduced in 2006 as part of the Microsoft .NET Framework 3.0. Although .NET 3.0 is automatically included in Windows Vista, you can also install it under Microsoft Windows XP with Service Pack 2 or Windows Server 2003 with Service Pack 1.

This book shows you how to write programs targeting the 3D graphics API of the Windows Presentation Foundation–or "WPF 3D," as it is known to its friends. This book is essentially a comprehensive examination of virtually all the classes and structures in the .NET namespace *System.Windows.Media.Media3D*, with plenty of code and markup examples.

The Role of WPF 3D

WPF 3D is *not* intended for graphics-intensive point-of-view games; nor is it suitable for producing the next big-screen epic featuring three-dimensional rodents or ogres. Programmers who want to pursue those types of applications might be happier looking at Microsoft DirectX rather than WPF 3D.

WPF 3D is instead intended to give programmers the ability to integrate 3D into their client Windows applications. This use of enhanced graphics might be as subtle as fashioning a control that has a 3D appearance, or using 3D to display complex information, or mimicking real-world objects (such as books). The last chapter of this book has some examples of WPF applications incorporating 3D that I hope will inspire you.

Although WPF 3D is not intended for complex games or movies, it is definitely built for animation. WPF includes an extensive animation API and you can use that API with your 3D graphics. In this book I begin demonstrating animation in Chapter 2, "Transforms and Animation," and I never let up. Somewhat related to animation is data binding. You can move or transform 3D figures by binding them to controls such as scrollbars–another of my favorite activities in this book.

Although WPF 3D runs on both Windows Vista and Windows XP with .NET 3.0 installed, you don't get exactly the same features. Even on Windows Vista, the quality of 3D graphics is dependent on the video board you have installed in the computer. A video board with a better on-board graphics processing unit (GPU) can accomplish some feats that are too slow to be done entirely in software. WPF graphics capabilities are categorized by "tiers" that are described on this Web page:

http://msdn2.microsoft.com/en-us/library/ms742196.aspx

In particular, only with a Tier 2 video board installed under Windows Vista do you get anti-aliasing in 3D. (Anti-aliasing is the use of shades of color to minimize the stark "staircase" effect caused by using discrete pixels to represent continuous lines or surfaces.) In the grand scheme of things, anti-aliasing might not sound like an important feature, but it makes a *big* difference when 3D graphics are animated.

You might want to get a new video board for your forays into 3D graphics, but if you're writing applications for other users, you might also want to be aware of the limitations that some of your users may experience when they run your programs.

Your Background

In writing this book, I have assumed that you already have experience programming for the Windows Presentation Foundation using the C# programming language and the Extensible Application Markup Language (XAML) that was introduced as part of .NET 3.0.

If you're a beginning programmer, I recommend that you learn C# first by writing console programs, which are character-mode programs that run in the Command Prompt window. My book *Programming in the Key of C#: A Primer for Aspiring Programmers* (Microsoft Press, 2003) takes this approach.

If you're a programmer who has a previous background in C or C++ but has not yet learned about programming for the .NET Framework with C#, you might want to begin with my short book *.NET Book Zero: What the C or C++ Programmer Needs to Know About C# and the .NET Framework*. The book is free and is available for reading or downloading from the following page of my Web site:

http://www.charlespetzold.com/dotnet

If you're familiar with earlier manifestations of .NET but haven't yet tackled .NET 3.0, WPF, and XAML, my book *Applications = Code + Markup: A Guide to the Microsoft Windows Presentation Foundation* (Microsoft Press, 2006) is a comprehensive tutorial. Several aspects of WPF programming are more crucial for 3D than others. These are:

- XAML syntax, including data binding and resources.
- Dependency properties.
- Animation and storyboards.
- Two-dimensional brushes.

If you are an experienced WPF programmer but you prefer to code in Microsoft Visual Basic .NET or another .NET-compliant language, I can only tell you that many of the programming examples in this book are in XAML rather than C#. Learning at least to read C# code and mentally translate it into your preferred language has become a vital skill in .NET programming.

Three-dimensional graphics programming necessarily involves mathematics, but I've tried to presume a minimum of background knowledge. For example, I've provided refreshers on vectors, matrix algebra, and imaginary numbers, but I've assumed that you have no previous knowledge of quaternions.

However, I do want you to come to this book with a basic facility with trigonometry. I don't need you to reel off lists of common trigonometric identities, but you should have a good working knowledge of angles, radians, sines, cosines, and tangents. If you know without thinking too hard that there are π radians in 180 degrees, that the sine of 90 degrees equals 1, that the cosine of zero degrees also equals 1, and that the tangent of 45 degrees equals 1 as well, you should be in good shape.

Some of the WPF 3D classes are specifically intended to insulate you from heavier mathematics going on under the covers. Consequently, I cover those classes early in the book. Not until relatively late in the book do I get into the more mathematics-laden topics of matrix transforms and quaternions. Depending on your ambitions and aspirations regarding 3D graphics programming, you might find these chapters challenging or altogether too scary. Books are great for conflicts like that because:

- You can skip something if you don't want to bother with it just now.
- You can come back and read something later—perhaps more than once.

My objective is to help you, not torture you.

System Requirements

To compile and run the programs described in this book, you'll need:

- Windows Vista, Windows XP with Service Pack 2, or Windows Server 2003 with Service Pack 1.
- The .NET Framework 3.0. This is included as part of Windows Vista; for Windows XP or Windows Server 2003 you can download it here:

  http://www.microsoft.com/downloads/details.aspx?familyid=10CC340B-F857-4A14-83F5-25634C3BF043
- The .NET Framework 3.0 Software Development Kit (SDK), available as a DVD image here:

  http://www.microsoft.com/downloads/details.aspx?familyid=7614FE22-8A64-4DFB-AA0C-DB53035F40A0

  or as a Web install here:

  http://www.microsoft.com/downloads/details.aspx?familyid=C2B1E300-F358-4523-B479-F53D234CDCCF

- Microsoft Visual Studio 2005 Standard Edition or Professional Edition.
- Visual Studio Extensions for .NET 3.0, available here:

  http://www.microsoft.com/downloads/details.aspx?familyid=F54F5537-CC86-4BF5-AE44-F5A1E805680D

In theory, you don't need Visual Studio to compile and run WPF applications. The .NET Framework 3.0 SDK includes a command-line program named MSBuild that builds WPF applications from C# project (.csproj) files. However, Visual Studio certainly makes WPF development easier.

The various links I've listed for downloading the .NET Framework, the SDK, and the Visual Studio extensions are all directly accessible from the 3D page of my Web site:

http://www.charlespetzold.com/3D

Go to the heading "Using the Book." Under that heading you'll also find a link to an Empty Project file for use with Visual Studio 2005, which is an approach to WPF programming that I prefer. (In fact, writing your own code rather than letting Visual Studio generate code for you is sometimes known as "Petzold style.")

At the time of this writing, the next version of Visual Studio (currently code-named Orcas) is available in a beta version. Orcas incorporates the .NET Framework 3.5 and the .NET Framework 3.5 SDK, and does not require any "extensions." As Orcas becomes more widely available, I'll have information about using it on the 3D page of my Web site.

For writing and experimenting with standalone XAML files, you can use XAMLPad, which is included with the SDK, or my own XamlCruncher, which you can install from the WPF page of my Web site:

http://www.charlespetzold.com/wpf

In particular, XamlCruncher 2.0 lets you load DLL files into the application domain. These files are then accessible to the XAML file you're developing.

Code Samples

All the code samples shown in this book (and some that are mentioned but not shown in these pages) can be downloaded from the book's companion content page maintained by Microsoft Press at the following Web site:

http://www.microsoft.com/mspress/companion/9780735623941

Purchase of this book gives you a royalty-free license to use any code samples (or modified code samples) you might find useful in your own programs, including commercial software.

(That's one of the purposes of this book.) However, you cannot republish the code samples. (That's why they're copyrighted.) Obviously I can't guarantee that the source code is applicable for specific purposes or even that it works right. (That's why it's free.)

If you page through this book, you'll notice that it has a number of pictures. Some of these are "screen shots" taken of the sample code running under Windows Vista. But others are diagrams and figures that help explain the concepts I'm discussing. All these other diagrams and figures were created with XAML files. The name of the particular XAML file is indicated in italics under each of these figures. You can find these XAML files in the Figures directory of the downloadable code. Running some of these XAML files requires loading the Petzold.Media3D library (which I'll describe shortly) into XamlCruncher 2.0.

Petzold.Media3D and Other Tools

The downloadable code for this book also includes source code for a dynamic-link library named Petzold.Media3D.dll that contains some classes that might be helpful in your 3D programming. If you're running XamlCruncher 2.0, you can load this DLL into the program's application domain and access it from XAML files that you create.

More recent versions of the Petzold.Media3D library are available for downloading from the 3D page of my Web site:

http://www.charlespetzold.com/3D

Purchase of this book gives you a royalty-free license to include this DLL with your own programs, including commercial software. You can also use any of the source code (including modified versions of the source code) in compilations of your own programs. However, I request that you do not distribute modified versions of the library itself. If you'd like to enhance the library in some way, do so by deriving from the classes in the library. I also ask that you do not distribute any of the source code that contributes to this library, either in a modified or unmodified state.

The Petzold.Media3D library is only one of several WPF 3D libraries available to the programmer. In particular, the WPF 3D team at Microsoft has put together a 3DTools library available here:

http://www.codeplex.com/3DTools

The WPF 3D team maintains a blog that often contains essential information here:

http://blogs.msdn.com/wpf3d

Support for This Book

Every effort has been made to ensure the accuracy of this book and the companion content. As corrections or changes are collected, they will be added to a Microsoft Knowledge Base article.

Microsoft Press provides support for books and companion content at the following Web site:

http://www.microsoft.com/learning/support/books

Questions and Comments

If you have comments, questions, or ideas regarding this book or the companion content, or if you have questions that are not answered by visiting the sites previously mentioned, please send them to Microsoft Press via e-mail at:

mspinput@microsoft.com

or via postal mail at:

Microsoft Press
Attn: 3D Programming for Windows *Editor*
One Microsoft Way
Redmond, WA 98052-6399

Please note that Microsoft software product support is not offered through these addresses.

Author's Web Site

Information specific to this book can be found on this page of my Web site:

http://www.charlespetzold.com/3D

Information about my other books, as well as a blog and miscellaneous articles, can be accessed from the home page of my Web site.

Special Thanks

Graphics programming in Windows has always especially appealed to me. In years gone by, I began writing books about Windows graphics programming—and even one about graphics programming for the OS/2 Presentation Manager—but something else always came up and these books were never completed. (Two covers of these abandoned books can be seen at the bottom of the Books page of my Web site.)

The prospect of writing a book about 3D graphics programming for Windows was very exciting. After I begged to write this book, my agent Claudette Moore and Microsoft Press Acquisitions Editor Ben Ryan helped make it reality. Thank you very much!

Apparently Project Editor Valerie Woolley and Technical Editor Kenn Scribner were sufficiently recovered from the experience of working with me on *Applications = Code + Markup* that we were able to reunite the "team" for this book. I am very thankful for their tireless work to help make this a book we can hold up with pride.

Positioned at the vanguard in the constant battle to prevent civilization from degenerating into chaos and brutality are copy editors. They help keep the English language clean from the evils of split infinitives, dangling participles, mismatched tenses, and the passive voice. I am forever grateful to the diligence of my copy editor Becka McKay in fixing my prose and helping me write with as much clarity as possible.

Eric Sink and Larry Smith graciously volunteered to read raw drafts of this book, and I am fortunate they did not also ask me to pay for the psychiatric counseling undoubtedly vital to their recovery from the experience. Their feedback and typo-detection skills were invaluable.

My friends at Microsoft continue to be generous with their knowledge and wisdom. From the 3D team I thank Daniel Lehenbauer, Jordan Parker, Adam Smith, Greg Schechter, and Peter Antal. I have also benefited greatly from the encouragement of Stephen Toub, Pablo Fernicola, Tim Sneath, and Paul Scholz.

In e-mails and blog entries, Larry O'Brien, Rob Hill, and Nathan Dunlap have given me advice and lessons. The inspiration for the StatePopulationAnimator program came from a discussion with Neil Devadasan.

And, of course, very much love and thanks go to Deirdre, who has helped make the past decade the very best years of my life.

Charles Petzold
New York City and Roscoe, New York
December 2006–June 2007

Contents at a Glance

Table of Contents

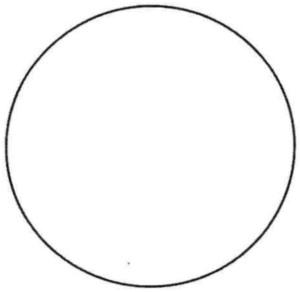
Chapter 1

Lights! Camera! Mesh Geometries!

In this chapter:

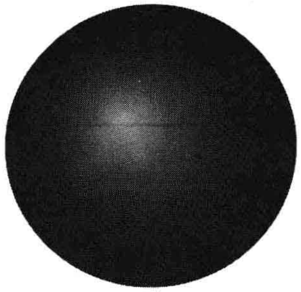
Ever since artists of the Upper Paleolithic era began adorning cave walls with images of hunters and their prey, people have strived to depict three-dimensional, real-world objects on two-dimensional surfaces. Our newspapers, magazines, books, museums, scratchpads, photo albums, movie theatres, video libraries, and computers are filled with the results.

Human perception is so attuned to the three dimensions of the real world that we are easily persuaded to accept even simple drawings as representing actual objects. This, for example, is obviously just a geometric circle:



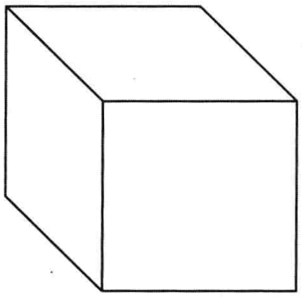
Circle.xaml

But add a little shading, and it becomes a ball:

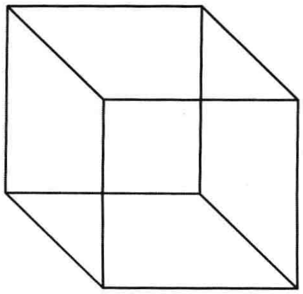

Ball.xaml

You don't need any 3D programming to display such a ball: The interior of a circle is simply colored with a *RadialGradientBrush*. The color varies from red on the edges (rendered as gray on the page) to white at the point specified by the *GradientOrigin* property. Set the *Gradient-Origin* off-center to suggest a light source from the upper left, which has become a convention in on-screen computer graphics.

The following object is depicted solely by its edges, and yet it is easily recognizable:


SolidCube.xaml

Although we can't see the entire object, only extreme skeptics would suggest that the back part of the object is much different from the front. An alternative version of the classic cube shows all the edges:


HollowCube.xaml