

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/325807486>

# Practical WPF Graphics Programming

Book · August 2007

---

CITATION

1

---

READS

13,155

1 author:



Ji-Hai Xu

[drxudotnet.com](http://drxudotnet.com)

43 PUBLICATIONS 754 CITATIONS

SEE PROFILE

# **Practical WPF Graphics Programming**

Advanced .NET Graphics Development with  
the Windows Presentation Foundation



# Practical WPF Graphics Programming

Advanced .NET Graphics Development with  
the Windows Presentation Foundation

*Jack Xu, Ph.D*



**UniCAD Publishing**

## **Practical WPF Graphics Programming**

Copyright © 2007 by Jack Xu, Ph.D

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

Editor: Anna Hsu

All rights reserved. No part of the contents of this book and corresponding example source code may be reproduced or transmitted in any form or by any means without the written permission of the publisher and the author.

The author and publisher have made every effort in the preparation of this book to ensure the accuracy of the information, however this book is sold without warranty, either express or implied. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained in the book.

The publisher offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and /or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact:

sales@unicadpublish.com

Visit us on the website: [www.unicadpublish.com](http://www.unicadpublish.com)

Published by UniCAD Publishing.

Phoenix, USA

ISBN-13: 978-0-9793725-1-3

ISBN-10: 0-9793725-1-8

### **Publisher's Cataloging-in-Publication Data**

Xu, Jack

Practical WPF Graphics Programming – Advanced .NET Graphics Development with the Windows Presentation Foundation / Jack Xu

– 1<sup>st</sup> ed.

p.cm.

ISBN 978-0-9793725-1-3

1. Windows Presentation Foundation. 2. Graphics Programming. 3. .NET Applications  
I. Title. II. Title III Title: Practical WPF Graphics Programming

*For my wonderful family*



# Contents

<b>Introduction .....</b>	<b>xvii</b>
Overview .....	xvii
What this Book Includes .....	xix
Is This Book for You?.....	xix
What Do You Need to Use This Book?.....	xx
How the Book Is Organized.....	xx
Using Code Examples .....	xxii
Customer Support .....	xxii
 <b>Chapter 1 Overview of WPF Programming.....</b>	<b>1</b>
New Features in WPF .....	1
XAML basics .....	2
Why XAML Needed? .....	2
Creating XAML Files.....	3
Code-Behind Files .....	4
Your first WPF Program .....	5
Properties in XAML .....	6
Event Handlers in Code-Behind Files .....	7
Code-Only Example.....	8
XAML-Only Example .....	10
 <b>Chapter 2 WPF Graphics Basics in 2D .....</b>	<b>13</b>
2D Coordinate Systems in WPF .....	13
Default Coordinates.....	13
Custom Coordinates .....	16
Custom Coordinates for 2D Charts .....	22
2D Viewport .....	27
Zooming and Panning.....	29



Basic 2D Graphics Shapes .....	31
Lines .....	31
Rectangles and Ellipses .....	33
Polylines .....	37
Polygons .....	38
<b>Chapter 3 2D Transformations .....</b>	<b>43</b>
Basics of Matrices and Transforms .....	43
Vectors and Points .....	44
Scaling .....	44
Reflection .....	45
Rotation .....	46
Translation .....	47
Homogeneous Coordinates .....	47
Translation in Homogeneous Coordinates .....	48
Scaling in Homogeneous Coordinates .....	48
Rotation in Homogeneous Coordinates .....	49
Combining Transforms .....	50
Vector and Matrix in WPF .....	51
Vector Structure .....	51
Matrix Structure .....	53
Matrix Operations .....	54
Matrix Transforms .....	57
Creating Perpendicular Lines .....	64
Object Transforms in WPF .....	69
MatrixTransform Class .....	71
ScaleTransform Class .....	75
TranslateTransform Class .....	79
RotateTransform Class .....	80
SkewTransform Class .....	81
Composite Transforms .....	83
<b>Chapter 4 Geometry and 2D Drawing .....</b>	<b>87</b>
Path and Geometry Classes .....	87
Line, Rectangle, and Ellipse Geometries .....	88
GeometryGroup Class .....	89
CombinedGeometry Class .....	92
PathGeometry Class .....	96
Lines and Polylines .....	97

Arcs.....	97
Bezier Curves.....	98
Geometry and Mini-Language .....	100
Interactive 2D Drawing.....	103
Mouse Events .....	114
Creating and Combining Shapes .....	115
Dragging and Moving Shapes .....	116
Hit Testing.....	117
Custom Shapes.....	121
Star Shape.....	121
Arrow Line .....	125
Testing Custom Shapes .....	131

## **Chapter 5 Colors and Brushes .....137**

Colors.....	137
System Colors .....	138
Color Picker.....	142
Brushes.....	150
SolidColorBrush.....	151
LinearGradientBrush .....	153
Custom Colormap Brush.....	158
RadialGradientBrush.....	165
Custom Radial Colormap Brush .....	169
ImageBrush .....	175
DrawingBrush .....	181
VisualBrush.....	187
Bitmap Effects.....	189
Opacity Masks.....	194
Brush Transforms.....	196
LinearGradientBrush Transform .....	204
RadialGradientBrush Transform .....	206
ImageBrush Transform .....	207
Drawing and Visual Brush Transform .....	207

## **Chapter 6 Animation .....209**

WPF Animation Basics.....	209
Property-Based Animation.....	210
A Simple Animation in Code .....	211
Animation and Storyboard.....	213

## x | Contents

Storyboard and Event Trigger .....	213
Storyboard Animation in Code .....	215
Animation and Timing Behavior.....	217
Duration .....	218
RepeatBehavior.....	220
AutoReverse .....	220
BeginTime .....	221
Speed Controls.....	221
Interactive Control.....	224
Animation and Transform.....	228
Animating Translation.....	228
Rolling Balls.....	231
Combining Transform Animations .....	236
Path Animation .....	240
Frame-Based Animation .....	244
Key-Frame Animation.....	244
Spline Key-Frame Animation .....	247
Custom Animation.....	249
Custom Animation Class.....	250
Per-Frame Animation .....	254
Animation Using Timer .....	256

## Chapter 7 Physics and Games in WPF .....263

Ordinary Differential Equations .....	263
Fourth-Order Runge-Kutta Method .....	264
Higher-Order ODEs .....	265
ODE Solver .....	265
Pendulum .....	267
Equation of Motion .....	267
Pendulum Simulator.....	268
Coupled-Spring System .....	274
Equations of Motion.....	275
Coupled Spring Simulator.....	275
Projectiles.....	287
Aerodynamic Drag Force .....	287
Projectile Equations of Motion .....	288
Golf Game .....	289
Collision.....	296
Bouncing Ball Physics .....	297

Bouncing Ball Simulator .....	299
Fractals .....	306
Binary Tree .....	307
Snowflake .....	310
Mandelbrot Set .....	315
Julia Set .....	323

## **Chapter 8 Charts in WPF .....331**

Simple Line Charts .....	331
Creating Simple Line Charts .....	332
How It Works .....	334
Line Charts with Data Collection .....	334
Chart Style .....	335
Data Collection .....	336
Data Series .....	337
Creating Charts .....	339
Gridlines and Labels .....	342
XAML Layout .....	342
ChartStyleGridlines Class .....	344
Testing Project .....	349
Legend .....	351
Legend Class .....	352
Testing Project .....	355
Chart User Control .....	358
Creating Chart Control .....	359
Defining Dependency Properties .....	361
Chart Style for Chart Control .....	366
Data Collection and Data Series .....	371
Using Chart Control .....	372
Creating Simple Chart .....	373
Creating Multiple Charts .....	375

## **Chapter 9 3D Transformations .....379**

3D Matrices in WPF .....	379
3D Points and Vectors .....	379
Matrix3D Structure .....	382
Matrix3D Operations .....	383
Matrix3D Transforms .....	387
Rotation and Quaternion .....	391

## xii | Contents

Projections.....	394
Orthographic Projections.....	395
Multi-View Projections .....	395
Axonometric Projections .....	396
Perspective Projections .....	399
One-Point Perspective Projections .....	400
Two-Point Perspective Projections.....	400
Three-Point Perspective Projections.....	401
Perspective Projection Matrix.....	401
Views and Projections in WPF .....	402
View Transform .....	403
Perspective Projection .....	406
View Frustum .....	406
Perspective Transform Matrix .....	406
Implementing Perspective Transforms .....	409
Testing Perspective Projections .....	411
Orthographic Projection .....	415
Orthographic Transform Matrix .....	416
Implementing Orthographic Transforms .....	417
Testing Orthographic Projections .....	418
Object Transforms in WPF .....	420
ScaleTransform3D Class .....	421
TranslateTransform3D class .....	425
RotateTransform3D Class .....	426
MatrixTransform3D Class.....	428
Combining Transforms .....	429

## **Chapter 10 WPF Graphics Basics in 3D .....433**

3D Graphics Basics.....	433
Viewport3D .....	434
3D Objects in WPF .....	435
Geometry and Mesh .....	436
GeometryModel3D and Surfaces .....	438
Illuminating the Scene.....	440
Camera Position .....	441
Simple Triangle in 3D .....	442
Basic 3D Shapes .....	444
Coordinate Axes and Wireframe .....	445
Creating Cube.....	447

Creating Sphere .....	453
Creating Cylinder .....	456
Creating Cone.....	461
Creating Torus.....	464

## **Chapter 11 Custom 3D Geometries.....469**

Cube Geometry .....	469
CubeGeometry Class .....	469
Testing CubeGeometry.....	473
Ellipsoid Geometry .....	477
EllipsoidGeometry Class .....	477
Testing EllipsoidGeometry .....	480
Cylindrical Geometry.....	482
CylinderGeometry Class .....	483
Testing CylinderGeometry .....	486
Cone Geometry .....	489
ConeGeometry Class .....	490
Testing ConeGeometry .....	493
Torus Geometry .....	497
TorusGeometry Class .....	497
Testing TorusGeometry.....	501
Icosahedron Geometry .....	505
IcosahedronGeometry Class.....	506
Testing IcosahedronGeometry .....	509
Dodecahedron Geometry .....	512
DodecahedronGeometry Class .....	513
Testing DodecahedronGeometry.....	519
Soccer Ball Geometry .....	521
SoccerGeometry Class .....	522
Testing SoccerGeometry .....	529

## **Chapter 12 Custom 3D Shapes .....535**

Deriving from UIElement3D .....	535
Cube Shape .....	538
Cube Shape Class .....	538
Testing Cube Shape.....	540
Ellipsoid Shape .....	544
Ellipsoid Shape Class .....	544
Testing Ellipsoid Shape.....	547

Cylinder Shape.....	548
Cylinder Shape Class .....	549
Testing Cylinder Shape .....	551
Cone Shape .....	552
Cone Shape Class .....	553
Testing Cone Shape.....	555
Torus Shape .....	556
Torus Shape Class .....	557
Testing Torus Shape.....	559
Icosahedron Shape .....	560
Icosahedron Shape Class.....	561
Testing Icosahedron Shape.....	563
Dodecahedron Shape .....	563
Dodecahedron Shape Class .....	563
Testing Dodecahedron Shape.....	564
Soccer Shape.....	565
Soccer Shape Class.....	565
Testing Soccer Shape .....	569
Combining Shape.....	569
Combining Shape Class .....	570
Testing Combining Shape .....	573

## **Chapter 13 3D Surfaces .....575**

Simple Surfaces .....	575
Rectangular Meshes .....	576
SimpleSurface Class.....	577
Creating Simple Surfaces .....	581
Parametric Surfaces .....	584
ParametricSurface Class.....	585
Creating Parametric Surfaces .....	589
Helicoid Surface .....	589
Sphere Surface .....	590
Torus Surface.....	591
Quadric Surfaces.....	592
Extruded Surfaces .....	593
ExtrudeSurface Class .....	594
Creating Extruded Surfaces.....	598
Surfaces of Revolution.....	599
RotateSurface Class.....	600

Creating Surfaces of Revolution .....	604
Surface Shading .....	606
SurfaceShading Class .....	606
Creating Shaded Surfaces.....	610

## **Chapter 14 3D Model Manipulation .....615**

Lighting and Shading.....	615
Light Sources.....	616
Testing Light Sources.....	617
Shading.....	620
Materials .....	621
Diffuse Materials.....	622
Specular and Emissive Materials .....	624
Texture Mapping.....	627
Using LinearGradientBrush.....	627
Using RadialGradientBrush.....	630
Using Image and Tile Brushes.....	633
2D Elements on 3D Surfaces .....	634
Viewport2DVisual3D Class .....	635
Using Viewport2DVisual3D .....	635
Interacting with 3D Models .....	641
Hit-Testing for 3D Geometries .....	642
Hit-Testing for 3D Shapes .....	645
Rotating 3D Objects with Mouse .....	649

## **Index .....653**





# Introduction

## Overview

Welcome to *Practical WPF Graphics Programming*. This book will provide all the tools you need to develop professional graphics applications using the Windows Presentation Foundation (WPF) and C# based on the .NET framework. I hope this book would be useful for WPF and C# programmers of all skill levels.

As a C# programmer, you are probably already familiar with Windows Forms, the mature and full-featured development tool. Windows Forms is built on top of the .NET framework and uses the Windows API to create the visual appearance of standard user interface elements. It provides all kinds of tools for laying out windows, menus, dialogs, and controls. You can also develop graphics applications based on Windows Forms using the Graphical Device Interface (GDI+). However, creating a feature-rich graphics application using Windows Forms can be a difficult and tedious task. For example, Windows Forms provides no tools for creating three-dimensional (3D) graphics applications. Even a 3D point, the simplest of 3D graphics objects, must be defined first in a suitable 3D coordinate system before it can be used as a 3D graphics object.

WPF completely changes the landscape for graphics programming. At first, you may think that WPF is just another way to create windows, menus, dialogs, and controls. However, WPF has much more to offer than any other Windows programming framework does. It integrates three basic Windows elements – text, controls, and graphics – into one single programming model, and puts these three elements into the same element tree in the same manner.

Without WPF, developing a graphics application might have involved using a number of different technologies, ranging from GDI/GDI+ for 2D graphics to Direct3D or OpenGL for 3D graphics. On the contrary, WPF is designed as a single model for graphics application development, providing seamless

## xviii | Introduction

integration between such services within an application. Similar constructs can be used for creating animation, data binding, and 3D models.

To take further advantage of new powerful graphics hardware technologies, WPF implements a vector-based graphics model. This allows for graphics to be scaled based on screen-specific resolution without the loss of image quality, something nearly impossible with fixed-size raster graphics. In addition, WPF leverages Direct3D for vector-based rendering and makes use of the graphics processing unit on any video card that implements DirectX in hardware.

With WPF, graphics elements can be easily integrated into any part of your user interface. For example, WPF provides 2D shape elements that can be involved in the user interface (UI) tree like other elements can. You are free to mix these shapes with any other kind of element, such as a button. The WPF 3D model is based on the Direct3D technology and allows you to create a custom 3D shape library that can be reused in your projects. The main benefits that WPF offers in 3D are its ease of use and its ability to integrate 3D content anywhere in a WPF application.

As you may have already noticed, there are many WPF programming books available in bookstores. The vast majority of these books are general-purpose user guides and tutorials, which explain the basics of WPF and how to use it to implement simple WPF applications. To take full advantage of WPF graphics features, however, there is a need for a book to provide an in-depth introduction specifically to WPF graphics programming.

This book is written with the intention of providing you with a complete and comprehensive explanation of the WPF graphics capability, and pays special attention to the code implementation details, which will be useful when you create your own real-world WPF graphics Applications. This book includes over 120 code examples, which cover broad array of topics on WPF graphics programming. Much of this book contains original work based on my own programming experience when I was developing commercial Computer Aided Design (CAD) packages. Without WPF and the .NET framework, developing advanced graphics is a difficult and time-consuming task. To add even simple charts or graphs to your applications, you often have to waste effort creating a chart program, or buy commercial graphics and chart add-on packages.

*Practical WPF Graphics Programming* provides everything you need to create advanced graphics in your WPF applications. It shows you how to create a variety of graphics, ranging from simple 2D shapes to complex 3D surfaces and interactive 3D models. I'll try my best to introduce you to WPF graphics programming in a simple way – simple enough to be easily followed by a beginner who has never had experience developing WPF graphics applications before. You can learn from this book how to create a full range of 2D and 3D graphics applications and how to implement custom 3D geometries and shapes that can be reused in your WPF projects.

## What this Book Includes

This book and its sample code listings, which are available for download at my website at [www.authors.unicadpublish.com/~jack\\_xu](http://www.authors.unicadpublish.com/~jack_xu), provide you with:

- A complete, in-depth instruction on practical WPF graphics programming. After reading this book and running the example programs, you will be able to add various sophisticated graphics to your WPF applications.
- Over 120 ready-to-run example programs that allow you to explore the graphics techniques described in the book. These examples can be used to better understand how graphics algorithms work. You can modify the code examples or add new features to them to form the basis of your own projects. Some of the example code listings provided in this book are already sophisticated graphics packages that can be used directly in your own real-world WPF applications.
- Many classes in the sample code listings that you will find useful in your WPF graphics programming. These classes contain matrix manipulation, coordinate transformation, color maps, chart controls, and the other useful utility classes. You can extract these classes and plug them into your own applications.

## Is This Book for You?

You don't have to be an experienced WPF developer or an expert to use this book. I designed this book to be useful to people of all levels of WPF programming experience. In fact, I believe that if you have some experience with the programming language C#, Windows Forms, HTML, and the .NET framework, you will be able to sit down in front of your computer, start up Microsoft Visual Studio 2008 and .NET 3.5, follow the examples provided in this book, and quickly become familiar with WPF graphics programming. For those of you who are already experienced WPF developers, I believe this book has much to offer as well. There is a great deal of information in this book about graphics programming not available in other WPF tutorial and reference books. In addition, most of the example programs in this book can be used directly in your own real-world application development. This book will provide you with a level of detail, explanation, instruction, and sample program code that will enable you to do just about anything WPF graphics-related.

The majority of the example programs in this book can be used routinely by WPF developers and technical professionals. Throughout the book, I'll emphasize the *usefulness* of WPF graphics programming to real-world applications. If you follow the instructions presented in this book closely, you'll be able to easily develop various practical WPF graphics applications, from 2D graphics and charts to a sophisticated 3D model libraries. At the same time, I'll not spend too much time discussing programming style, execution speed, and code optimization, because there is a plethora of books out there that already

## xx | Introduction

deal with such topics. Most of the example programs you'll find in this book omit error handlings. This makes the code easier to understand by focusing only on the key concepts and practical applications.

## What Do You Need to Use This Book?

You'll need no special equipment to make the best use of this book and understand the algorithms. To run and modify the sample programs, you'll need a computer that is capable of running either Windows Vista or Windows XP. The software installed on your computer should include Visual Studio 2008 and the .NET 3.5 standard edition or higher. If you have Visual Studio 2005 and .NET 3.0, you can also run most of the sample code with few modification. Please remember, however, that this book is intended for Visual Studio 2008 and .NET 3.5, and that all of the example programs were created and tested on this platform, so it is best to run the sample code on the same platform.

## How the Book Is Organized

This book is organized into fourteen chapters, each of which covers a different topic about WPF graphics programming. The following summaries of each chapter should give you an overview of the book's content:

### Chapter 1, *Overview of WPF Programming*

This chapter introduces the basics of WPF and reviews some of the general aspects of WPF programming, including XAML files used to define user interfaces.

### Chapter 2, *WPF Graphics Basics in 2D*

This chapter reviews some fundamental concepts of the 2D graphics and the 2D drawing model in WPF. It introduces coordinate systems and basic 2D shapes.

### Chapter 3, *2D Transformations*

This chapter covers the mathematical basics for 2D graphics programming. 2D vectors, matrices, and transformations in the homogeneous coordinate system, including translation, scaling, reflection, and rotation, are discussed. These 2D matrices and transformations allow WPF applications to perform a wide variety of graphical operations on graphics objects in a simple and consistent manner.

### Chapter 4, *Geometry and 2D Drawing*

This chapter introduces WPF's Geometry classes and demonstrates why you need them to create complex 2D graphics objects. It also shows you how to create interactive 2D drawing programs and custom shapes.

### Chapter 5, *Colors and Brushes*

This chapter covers the color system and brushes that WPF uses to paint graphics objects. It introduces a variety of brushes and their transformations. You'll learn how to create exotic visual effects using different brushes, including the gradient, tile, and image brushes.

#### Chapter 6, *Animation*

This chapter describes WPF animation facilities, which allow most of the properties and transformations of the graphics objects (such as position, size, translation, rotation, etc.) to be animated. It also describes how to create a custom animation class that can be used in physics-based animation.

#### Chapter 7, *Physics and Games in WPF*

This chapter covers topics related to real-world WPF applications. You'll learn how to create and simulate physics models by solving ordinary differential equations with the Runge-Kutta method, and how to incorporate physics models into real-world games in WPF. This chapter discusses several physics models and games, including a pendulum, a coupled spring system, a golf ball (projectiles), ball collision, and fractals.

#### Chapter 8, *Charts in WPF*

This chapter contains instructions on creating 2D line charts in WPF. It introduces basic chart elements including the chart canvas, text canvas, axes, title, labels, ticks, and legend. From this chapter, you'll also learn how to put a 2D chart application into a custom user control and how to reuse this control in your WPF applications.

#### Chapter 9, *3D Transformations*

This chapter extends the concepts described in Chapter 3 into the third dimension. It explains how to define 3D graphics objects and how to translate, scale, reflect, and rotate these 3D objects. It also describes transformation matrices that represent projection and transformations, which allow you to view 3D graphics objects on a 2D screen. You'll also learn how WPF defines 3D vectors, matrices, and projections.

#### Chapter 10, *WPF Graphics Basics in 3D*

This chapter explores the basics of 3D models in WPF. It introduces Viewport3D, the 3D geometry and the mesh model, lighting, camera, etc. You'll also learn how to create basic 3D shapes directly in WPF.

#### Chapter 11, *Custom 3D Geometries*

This chapter explains how to create custom geometries for various 3D shapes. These custom geometry classes can be used as resources in XAML files, and these resources can be used in your markup with a data binding.

#### Chapter 12, *Custom 3D Shapes*

This chapter shows how to implement custom 3D shape classes. Unlike custom 3D geometry classes, which can only be used as shareable resources, these

## xxii | Introduction

custom 3D shape classes can be used directly in your XAML files in the same way as the 2D shapes, such as the Line, Rectangle, and Ellipse. You can use these custom 3D shape classes to create a powerful 3D model library.

### Chapter 13, *3D Surfaces*

This chapter explains how to create various 3D surfaces, from simple surfaces to complex surfaces, using rectangular meshes and different techniques, including parametric, extrusion, and revolution approaches. It also describes how to add lighting and shading effects to these surfaces.

### Chapter 14, *3D Model Manipulation*

This chapter covers broad array of topics on manipulating 3D models in WPF. It describes how to create various 3D special effects using different materials, different light sources, and texture maps. It also explains how to place interactive 2D elements on 3D surfaces and how to rotate 3D graphics objects with the mouse using the virtual trackball method.

## Using Code Examples

You may use the code in this book in your applications and documentation. You don't need to contact the author or the publisher for permission unless you are reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book doesn't require permission. Selling or distributing the example code listings does require permission. Incorporating a significant amount of example code from this book into your applications and documentation also requires permission. Integrating the example code from this book into commercial products isn't allowed without written permission of the author.

## Customer Support

I am always interested in hearing from readers, and would like to hear your thoughts on this book. You can send me comments by e-mail to [jxu@drxudotnet.com](mailto:jxu@drxudotnet.com). I also provide updates, bug fixes, and ongoing support via my website:

<http://drxudotnet.com>

You can also obtain the complete source code for all of examples in this book from the above website.