Quartz 2D Programming Guide

Developer

Contents

Introduction 13

Who Should Read This Document? 13

Organization of This Document 13

See Also 15

Overview of Quartz 2D 16

The Page 16

Drawing Destinations: The Graphics Context 17

Quartz 2D Opaque Data Types 19

Graphics States 21

Quartz 2D Coordinate Systems 22

Memory Management: Object Ownership 25

Graphics Contexts 26

Drawing to a View Graphics Context in iOS 26

Creating a Window Graphics Context in Mac OS X 27

Creating a PDF Graphics Context 29

Creating a Bitmap Graphics Context 34

- Supported Pixel Formats 39
- Anti-Aliasing 40

Obtaining a Graphics Context for Printing 41

Paths 42

Path Creation and Path Painting 42

The Building Blocks 44

- Points 44
- Lines 44
- Arcs 45
- Curves 46
- Closing a Subpath 48
- Ellipses 49
- Rectangles 49

Creating a Path 50

Painting a Path 51

2010-11-19 | © 2001, 2010 Apple Inc. All Rights Reserved.

Parameters That Affect Stroking 52
Functions for Stroking a Path 54
Filling a Path 55
Setting Blend Modes 56
Clipping to a Path 66

Color and Color Spaces 68
About Color and Color Spaces 68
The Alpha Value 69
Creating Color Spaces 71
Creating Device-Independent Color Spaces 71
Creating Generic Color Spaces 72
Creating Device Color Spaces 72
Creating Indexed and Pattern Color Spaces 73
Setting and Creating Colors 73
Setting Rendering Intent 75

Transforms 76
About Quartz Transformation Functions 77
Modifying the Current Transformation Matrix 77
Creating Affine Transforms 83
Evaluating Affine Transforms 85
Getting the User to Device Space Transform 85
The Math Behind the Matrices 85

Patterns 89
The Anatomy of a Pattern 89
Colored Patterns and Stencil (Uncolored) Patterns 93
Tiling 94
How Patterns Work 94
Painting Colored Patterns 95
Write a Callback Function That Draws a Colored Pattern Cell 95
Set Up the Colored Pattern Color Space 97
Set Up the Anatomy of the Colored Pattern 97
Specify the Colored Pattern as a Fill or Stroke Pattern 99
Draw With the Colored Pattern 99
A Complete Colored Pattern Painting Function 99
Painting Stencil Patterns 101
Write a Callback Function That Draws a Stencil Pattern Cell 102
Set Up the Stencil Pattern Color Space 103

- Set Up the Anatomy of the Stencil Pattern 104
- Specify the Stencil Pattern as a Fill or Stroke Pattern 104
- Drawing with the Stencil Pattern 104
- A Complete Stencil Pattern Painting Function 104

Shadows 107

How Shadows Work 108

Shadow Drawing Conventions Vary Based on the Context 108

Painting with Shadows 108

Gradients 112

Axial and Radial Gradient Examples 112

A Comparison of CGShading and CGGradient Objects 115

Extending Color Beyond the End of a Gradient 116

Using a CGGradient Object 118

Using a CGShading Object 121

- Painting an Axial Gradient Using a CGShading Object 123
- Painting a Radial Gradient Using a CGShading Object 130

See Also 135

Transparency Layers 136

How Transparency Layers Work 137

Painting to a Transparency Layer 138

Data Management in Quartz 2D 140

Moving Data into Quartz 2D 141

Moving Data out of Quartz 2D 143

Moving Data Between Quartz 2D and Core Image in Mac OS X 144

Bitmap Images and Image Masks 146

About Bitmap Images and Image Masks 147

Bitmap Image Information 147

- Decode Array 148
- Pixel Format 148
- Color Spaces and Bitmap Layout 149

Creating Images 151

- Creating an Image From Part of a Larger Image 152
- Creating an Image from a Bitmap Graphics Context 153

Creating an Image Mask 154

Masking Images 155

- Masking an Image with an Image Mask 155
- Masking an Image with an Image 157
- Masking an Image with Color 158
- Masking an Image by Clipping the Context 162

Using Blend Modes with Images 164

- Normal Blend Mode 165
- Multiply Blend Mode 166
- Screen Blend Mode 167
- Overlay Blend Mode 167
- Darken Blend Mode 168
- Lighten Blend Mode 169
- Color Dodge Blend Mode 170
- Color Burn Blend Mode 170
- Soft Light Blend Mode 171
- Hard Light Blend Mode 172
- Difference Blend Mode 173
- Exclusion Blend Mode 174
- Hue Blend Mode 175
- Saturation Blend Mode 176
- Color Blend Mode 177
- Luminosity Blend Mode 177

Core Graphics Layer Drawing 179

How Layer Drawing Works 180

Drawing with a Layer 181

- Create a CGLayer Object Initialized with an Existing Graphics Context 182
- Get a Graphics Context for the Layer 182
- Draw to the CGLayer Graphics Context 182
- Draw the Layer to the Destination Graphics Context 183

Example: Using Multiple CGLayer Objects to Draw a Flag 184

PDF Document Creation, Viewing, and Transforming 190

Opening and Viewing a PDF 191

Creating a Transform for a PDF Page 193

Creating a PDF File 196

Adding Links 198

Protecting PDF Content 198

PDF Document Parsing 200

Inspecting PDF Document Structure 200

Parsing PDF Content 202
- Write Callbacks for Operators 203
- Create and Set Up the Operator Table 204
- Open the PDF Document 205
- Scan the Content Stream for Each Page 206

PostScript Conversion 208
Writing Callbacks 208
Filling In a Callbacks Structure 209
Creating a PostScript Converter Object 210
Creating Data Provider and Data Consumer Objects 210
Performing the Conversion 211

Text 212
How Quartz 2D Draws Text 213
Controlling How Text Looks 213
Drawing Text 215
Measuring Text Before Drawing 219
Copying Font Variations 219
PostScript Fonts 220

Glossary 221

Document Revision History 224

Figures, Tables, and Listings

Overview of Quartz 2D

Overview of Quartz 2D 16
Figure 1-1 The painter's model 17
Figure 1-2 Quartz drawing destinations 18
Figure 1-3 Opaque data types are the basis of drawing primitives in Quartz 2D 20
Figure 1-4 The Quartz coordinate system 22
Figure 1-5 Modifying the coordinate system creates a mirrored image. 24
Table 1-1 Parameters that are associated with the graphics state 21

Graphics Contexts

Graphics Contexts 26
Figure 2-1 A view in the Cocoa framework that contains Quartz drawing 27
Figure 2-2 A PDF created by using CGPDFContextCreateWithURL 29
Figure 2-3 An image created from a bitmap graphics context and drawn to a window graphics context 38
Figure 2-4 A comparison of aliased and anti-aliasing drawing 40
Table 2-1 Pixel formats supported for bitmap graphics contexts 39
Listing 2-1 Drawing to a window graphics context 28
Listing 2-2 Calling `CGPDFContextCreateWithURL` to create a PDF graphics context 30
Listing 2-3 Calling `CGPDFContextCreate` to create a PDF graphics context 31
Listing 2-4 Drawing to a PDF graphics context 32
Listing 2-5 Creating a bitmap graphics context 35
Listing 2-6 Drawing to a bitmap graphics context 37

Paths

Paths 42
Figure 3-1 Quartz supports path-based drawing 42
Figure 3-2 A path that contains two shapes, or subpaths 43
Figure 3-3 A clipping area constrains drawing 43
Figure 3-4 Multiple paths; each path contains a randomly generated circle 45
Figure 3-5 Defining an arc with two tangent lines and a radius 46
Figure 3-6 Multiple paths; each path contains a randomly generated curve 47
Figure 3-7 A cubic Bézier curve uses two control points 47
Figure 3-8 A quadratic Bézier curve uses one control point 48
Figure 3-9 Multiple paths; each path contains a randomly generated ellipse 49
Figure 3-10 Multiple paths; each path contains a randomly generated rectangle 50
Figure 3-11 Examples of line dash patterns 54

Figure 3-12 Concentric circles filled using different fill rules 56
Figure 3-13 The rectangles painted in the foreground 57
Figure 3-14 The rectangles painted in the background 57
Figure 3-15 Rectangles painted using normal blend mode 58
Figure 3-16 Rectangles painted using multiply blend mode 58
Figure 3-17 Rectangles painted using screen blend mode 59
Figure 3-18 Rectangles painted using overlay blend mode 59
Figure 3-19 Rectangles painted using darken blend mode 60
Figure 3-20 Rectangles painted using lighten blend mode 60
Figure 3-21 Rectangles painted using color dodge blend mode 61
Figure 3-22 Rectangles painted using color burn blend mode 61
Figure 3-23 Rectangles painted using soft light blend mode 62
Figure 3-24 Rectangles painted using hard light blend mode 62
Figure 3-25 Rectangles painted using difference blend mode 63
Figure 3-26 Rectangles painted using exclusion blend mode 63
Figure 3-27 Rectangles painted using hue blend mode 64
Figure 3-28 Rectangles painted using saturation blend mode 64
Figure 3-29 Rectangles painted using color blend mode 65
Figure 3-30 Rectangles painted using luminosity blend mode 65
Table 3-1 Parameters that affect how Quartz strokes the current path 52
Table 3-2 Line join styles 52
Table 3-3 Line cap styles 53
Table 3-4 Functions that stroke paths 54
Table 3-5 Functions that fill paths 56
Table 3-6 Functions that clip the graphics context 66
Listing 3-1 Setting up a circular clip area 66

Color and Color Spaces 68

Figure 4-1 Applying a BGR and an RGB color profile to the same image 69
Figure 4-2 A comparison of large rectangles painted using various alpha values 69
Figure 4-3 A comparison of global alpha values 70
Figure 4-4 A CMYK fill color and an RGB stroke color 73
Table 4-1 Color values in different color spaces 68
Table 4-2 Color-setting functions 74

Transforms 76

Figure 5-1 Applying scaling and rotation 77
Figure 5-2 An image that is not transformed 78
Figure 5-3 A translated image 78
Figure 5-4 A rotated image 79

Figure 5-5 A scaled image 80
Figure 5-6 An image that is translated and rotated 81
Figure 5-7 An image that is translated, scaled, and then rotated 82
Figure 5-8 An image that is rotated, scaled, and then translated 83
Table 5-1 Affine transform functions for translation, rotation, and scaling 84

Patterns 89
Figure 6-1 A pattern drawn to a window 89
Figure 6-2 A pattern cell 89
Figure 6-3 Pattern cells with black rectangles drawn to show the bounds of each cell 90
Figure 6-4 Spacing between pattern cells 90
Figure 6-5 A scaled pattern cell 92
Figure 6-6 A rotated pattern cell 92
Figure 6-7 A translated pattern cell 93
Figure 6-8 A colored pattern has inherent color 93
Figure 6-9 A stencil pattern does not have inherent color 93
Figure 6-10 A stencil pattern cell 102
Listing 6-1 A drawing callback that draws a colored pattern cell 96
Listing 6-2 Creating a base pattern color space 97
Listing 6-3 The CGPatternCreate function prototype 97
Listing 6-4 A function that paints a colored pattern 99
Listing 6-5 A drawing callback that draws a stencil pattern cell 102
Listing 6-6 Code that creates a pattern color space for a stencil pattern 103
Listing 6-7 Code that sets opacity for a colored pattern 104
Listing 6-8 A function that paints a stencil pattern 105

Shadows 107
Figure 7-1 A shadow 107
Figure 7-2 A shadow with no blur and another with a soft edge 107
Figure 7-3 A colored shadow and a gray shadow 109
Listing 7-1 A function that sets up shadows 109

Gradients 112
Figure 8-1 An axial gradient along a 45 degree axis 112
Figure 8-2 An axial gradient created with seven locations and colors 113
Figure 8-3 A radial gradient that varies between two circles 113
Figure 8-4 A radial gradient created by varying only the alpha component 114
Figure 8-5 A radial gradient that varies between a point and a circle 114
Figure 8-6 Nested radial gradients 115
Figure 8-7 Extending an axial gradient 117

Figure 8-8 Extending a radial gradient 117
Figure 8-9 A radial gradient painted using a CGGradient object 120
Figure 8-10 An axial gradient with three locations 121
Figure 8-11 An axial gradient that is clipped and painted 123
Figure 8-12 A radial gradient creating using a CGShading object 130
Table 8-1 Differences between CGShading and CGGradient objects 116
Listing 8-1 Creating a CGGradient object 118
Listing 8-2 Painting an axial gradient using a CGGradient object 119
Listing 8-3 Painting a radial gradient using a CGGradient object 119
Listing 8-4 The variables used to create a radial gradient by varying alpha 120
Listing 8-5 The variables used to create a gray gradient 121
Listing 8-6 Computing color component values 124
Listing 8-7 Creating a CGFunction object 125
Listing 8-8 Creating a CGShading object for an axial gradient 126
Listing 8-9 Adding a semicircle clip to the graphics context 126
Listing 8-10 Releasing objects 127
Listing 8-11 Painting an axial gradient using a CGShading object 127
Listing 8-12 Computing color component values 131
Listing 8-13 Creating a CGShading object for a radial gradient 131
Listing 8-14 Code that releases objects 132
Listing 8-15 A routine that paints a radial gradient using a CGShading object 132

Transparency Layers 136
Figure 9-1 Three circles as a composite in a transparency layer 136
Figure 9-2 Three circles painted as separate entities 137
Figure 9-3 Three rectangles painted to a transparency layer 138
Listing 9-1 Painting to a transparency layer 138

Data Management in Quartz 2D 140
Figure 10-1 Moving data to and from Quartz 2D in Mac OS X 141
Table 10-1 Functions that move data into Quartz 2D 142
Table 10-2 Functions that move data out of Quartz 2D 143

Bitmap Images and Image Masks 146
Figure 11-1 Bitmap images 147
Figure 11-2 32-bit and 16-bit pixel formats for CMYK and RGB color spaces in Quartz 2D 150
Figure 11-3 A subimage created from a larger image 152
Figure 11-4 An image, a subimage taken from it and drawn so it's enlarged 153
Figure 11-5 The original image 156
Figure 11-6 An image mask 156