
Index

A

Abstract Window Toolkit (AWT), 12–13

Acceleration, interpolating, 360–363

ActionTrigger, 384

Acyclic gradients, 195–197

Add composite, 171–175

Adobe kuler, 547

Aerith

- blending modes, 542

- coding applications

 - blending modes, 542

 - layers, 540–542

 - from screen mockups, 540–544

- history of, 531–532

- online resources, 550

- running, 532–533

- screen mockups

 - coding from, 540–544

 - colors, 545–547

 - gradients, 547

 - guides, 543

 - required skills, 544–545

 - workflow design, 538–539

- workflow design

 - with a graphics editor, 535–537

 - on paper, 533–535, 537–538

 - screen mockups, 538–543

Affine transforms, 63–70, 203–204

AffineTransformOp filter, 203–204

AJAX (Asynchronous JavaScript and XML), 2–3

AlphaComposite. *See also* Composites.

- antialiased-clipping, 168

- Clear rule, 157, 165

- common uses, 164–168

- creating, 163–164

- cutouts, 167

- description, 153–155

- destination pixels, 156

- drop shadow, 167

- Dst rule, 157

- DstAtop rule, 158

- DstIn rule, 158

- DstOut rule, 159

- DstOver rule, 159

- erasing a background, 157, 165

- picture frames, 167

- Porter-Duff equations, 153–154

- Porter-Duff rules, 155–162

- problems with, 168–170

- replacing existing drawings, 161, 166–168

- setting up, 163–164

- soft-clipping, 168

- source pixels, 156

- Src rule, 160

- SrcAtop rule, 160

- SrcIn rule, 161, 166–168

- SrcOut rule, 161

- SrcOver rule, 162, 165–166

- temporary offscreen images, 170

- translucency effect, 162, 165–166

- Xor rule, 162

AlphaComposite class, 468–470

Anchor points, 369

Animated Transitions, 497–530

Animation. *See also* Motion effects.

- frame rate

 - frame-based animation, 268

 - minimum required, 289–290

 - smoothing, 318, 332

Animation, *continued*

frame-based

flicker, 269

frame rate, 268

intermediate steps, 270

linear interpolation, 273–274

overview, 266–268

problems, 268–275

realistic motion, 272–275

speed, 269–271

Swing buffering, 269, 271–272

GUIs, 303–304

repeating, 345, 351–352

running, querying, 347

smoothing

antialiasing, 327, 340–341

apparent smoothness, 320

blur effect, 339–340

bouncing motion, 320

color effect, 320–329, 332, 338–339

color spaces, 322

consistency, 318–319

contrast, 325–326

demo, 335–341

double buffering, 334

frame rates, 318, 332

hard edges, 326–327

jumpy motion, 328–329

linear shapes, 332

motion blur, 328–329

object color, *versus* background,
325–326

perceived performance, 316

performance, 317–318

pixel color change, 322–323

realistic motion, 319–320

rendering, 337–341

resolution, optimizing, 318–319

straight edges, 327–328

tearing, 330

timing effect, 317–320

vertical retrace effect, 329–334

starting, 345, 347, 352–353

stopping, 347, 353

time-based

definition, 265

flicker, 269

intermediate steps, 270

linear interpolation, 273–274

problems, 268–275

realistic motion, 272–275

speed, 269–271

Swing buffering, 269, 271–272

timer resolution. *See also* Timing

Framework.

`currentTimeMillis()`, 291–293

definition, 288–289

increasing, 278

measuring, 297–299

milliseconds, 276–278, 291–293

nanoseconds, 278, 291–293

`nanoTime()`, 291–293

performance, 288, 290, 299–300

sleeping, 293–297

timer utilities. *See also* Timing

Framework.

callbacks, 284–285

`currentTimeMillis()`, 276–278

demos, 282–284, 286–288

fixed delay, 285

fixed rate, 285

interframe volatility, 277–278

milliseconds, 276–278

nanoseconds, 278

`nanoTime()`, 278

querying current time, 275–278

`sleep()`, 279–280, 293–297

sleeping, 279–282, 293–297

`Timer (java.util)`, 282`Timer (javax.swing)`, 284–288

timers, 280–284

`wait()`, 279–280, 295–297

wake-up calls, 279–282

transitions. *See* Transitions.

Animator class

`cancel()`, 347

constructor methods, 346

control flow, 347

- definition, 345
- description, 346
- `isRunning()`, 347
- `pause()`, 347
- `resume()`, 347
- `start()`, 347
- `stop()`, 347
- Antialiased-clipping, 168
- Antialiasing
 - demo, 56
 - description, 54–56
 - jaggies, 54
 - shapes, 57
 - smoothing animation, 327, 340–341
 - text, 57–59
- `AntiAliasingDemo` demo, 56
- Area averaging, image scaling, 103–104
- Articles. *See* Books and publications.
- Asynchronous JavaScript and XML (AJAX), 2–3
- Asynchronous paint requests, 17–19
- Authors' sites, xxiii, 551
- Autoboxing, 405
- Auto-reversing triggers, 381
- AWT (Abstract Window Toolkit), 12–13

- B**
- Back buffers, 29
- Backgrounds
 - color, graphics state, 49, 50
 - color, *versus* object color, 325–326
 - erasing, 157, 165
- `BasicRace` demo, 358–359, 363–364
- Bicubic image scaling, 102–104
- Bilinear image scaling, 102–104
- Blending modes, 542. *See also*
 - `AlphaComposite`; `Composites`.
- Blocking EDT, 31–32
- Bloom effect, 6, 478–482
- Blur
 - animation, 339–340
 - bokeh, 423–424
 - box, 426–428
 - depth of field, 423
 - description, 423–425
 - Gaussian, 428–432
 - motion blur, 328–329
 - performance, 433–434
 - reflection, 435–437
 - simple, 426–428
- Bokeh, 423–424
- Books and publications. *See also* Online resources.
 - “Compositing Digital Images,” 153
 - Computer Graphics Principles and Practices*, 377
 - Filthy Rich Clients*, online resources, 551
 - The Geometry Toolbox for Graphics and Modeling*, 377
 - “The Perils of `Image.getScaledInstance()`,” 98
 - System Properties for Java 2D...*, 145
 - user interface design, 547
 - visual design, 547
- Borders, on components, 75
- Boring stuff, not found
- Box blur, 426–428
- Bright spots, simulating, 189–192
- Brightening, 213–214, 444–446
- `brighter()`, 444
- `BufferedImage` objects
 - advantages of, 95–96
 - converting images to, 96–97
 - definition, 93
 - transparent/translucent images, 97
- `BufferedImageOp` filter, 201–203, 214–221
- Buttons layer, 238

- C**
- Caching
 - gradients, 193–195
 - intermediate images, 134–141
- Callbacks
 - cleanup operations, 348–349
 - definition, 345

Callbacks, *continued*

- Swing event thread, 284–285
 - timing events, receiving, 348–350
 - timing targets, 349–350
- Campbell, Chris, 98, 111, 147
- `cancel()`, 347
- Canceling triggers, 381
- Choppy animation. *See* Animation, smoothing.
- Circular gradients, 189–192
- Clear rule, 157, 165
- `clip()`, 61
- The clip
- constraining primitives to, 120
 - graphics state, 49, 61–62
 - honoring, 117–120
 - performance, 115–121
- Clipping images
- antialiased-clipping, 168
 - the clip, 121
 - soft-clipping, 168
- `clipRect()`, 61
- Coalescing paint requests, 18
- Color Schemer Online, 546
- Color Schemer Studio, 546
- Color spaces, animation, 322
- `ColorConvertOp` filter, 204–206
- Colors
- channels, 154
 - effects on animation, 320–329, 332, 338–339
 - fading, 470–472
 - graphics primitives, 63–64
 - mapping source/destination, 211–213
 - model conversion, 204–206, 211–213
 - object, *versus* background, 325–326
 - online resources, 546–547
 - opacity. *See* Opacity; Translucency; Transparency.
 - rules for. *See* AlphaComposite; Composites.
 - screen mockups, 545–547
- COLOURlovers, 547
- Combo-box drop-down layer, 238
- Command-line flags, 145–149

Compatible images

- converting to, 124–126
 - definition, 94
 - performance, 121–126
 - uses for, 122–123
- `compose()`, 174–177
- Composing pixels, 174–177
- `CompositeContext` composite, 174–175
- `CompositeEffect` effect, 512–513
- Composites. *See also* AlphaComposite.
- adding source/destination values, 171–174
 - composing pixels, 174–177
 - creating
 - Add composite, 171–175
 - `CompositeContext` composite, 174–175
 - overview, 170–171
 - defining area for, 174–177
 - graphics state, 49
 - immutability, 173
 - library of, 171
- “Compositing Digital Images,” 153
- Computer Graphics Principles and Practices*, 377
- Consistency, animation, 318–319
- Contrast, animation, 325–326
- Control points, 369
- Convolution kernel, 206–209
- `ConvolveOp` filter, 206–211
- Convolving images, 206–211
- Cooler applications, 1–552
- `copyArea()`, 83
- `CopyAreaPerformance` demo, 83–87
- Copying areas, 83–87
- Cross-fading, 472
- `currentTimeMillis()`, 276–278, 291–293
- Custom interpolation, 377–378
- Cutouts, 167
- Cyclic gradients, 195–197

D

- Darkening images, 213–214
- Data binding, 1

- DataBuffer, grabbing, 129–132
- Deadlock, 35–37
- Debugging performance, 148–149
- Deceleration, interpolating, 360–363
- Demos
 - animation
 - FadingButton, 304–306
 - MovingButton, 306–310
 - MovingButtonContainer, 310–314
 - SmoothMoves, 335–341
 - SwingTimerDemo, 286–288
 - UtilTimerDemo, 282–284
 - AntiAliasingDemo, 56
 - color, graphics primitives, 63–64
 - CopyAreaPerformance, 83–87
 - copying areas, 83–87
 - custom components, matching desktop settings, 59–68
 - DiagonalLine, 74–75
 - donut shape, 79
 - drawing, 87–89
 - drawing shapes, 80–83
 - DrawShapes, 80–83
 - Fading, 468–470
 - FillDraw, 87–89
 - filling, 87–89
 - filling a blank canvas, 22–23
 - FontHints, 59–68
 - gradients, 180–181
 - graphics state
 - persistence, 71–72
 - setting, 51–54
 - image scaling
 - high resolution pictures, 109–111
 - photographs, 105–109
 - progressively larger images, 105–109
 - scale(), 63–64, 67–68
 - vector drawings, 105–109
 - ImageBrowser, 519–526
 - IntermediateImages, 137–142
 - loaded file names, displaying, 39–41
 - loading images from a hard drive, 39–41
 - Morphing, 491–495
 - performance, 137–142
 - PictureScaler, 109–111
 - Pulse, 474–478
 - PulseField, 482–484
 - reflection, 253–262
 - RepaintManager, 253–262
 - rotating objects, 63–64, 67–68, 68–74
 - RotationAboutCenter, 68–74
 - ScaleTest, 105–109
 - scaling images, 105–109, 109–111
 - SearchTransition, 503–509, 516–518
 - simulated depth, 180–181
 - SmoothMoves, 335–341
 - Spring, 486–489
 - text hints, setting, 59–68
 - Timing Framework
 - BasicRace, 358–359, 363–364
 - bouncing spheres, 382–392
 - FadingButton, 353–356
 - interpolation, 363–364
 - Interpolator, 372–378
 - interpolator race, 363–364
 - KeyFrames class, 414–419
 - MultiStepRace, 414–419
 - property setters, 393–395
 - PropertySetter class, 398
 - racetrack, 356–359, 398–399, 414–419
 - SetterRace, 398
 - spline interpolation, 372–375, 376–378
 - SplineEditor, 376–378
 - SplineInterpolatorTest, 372–375
 - triggers, 382–392
 - transformations, 63–64, 67–68
 - translating objects, 63–64, 67–68
 - translucency, 249–251
 - TranslucentPanel, 249–251
- Depth of field, 423
- derive(), 164
- Design books, 547
- Design tools. *See* Aerith.
- Desktop Java Graphics APIs. *See* AWT (Abstract Window Toolkit); Java 2D; Swing.
- Destination pixels, 156

- DiagonalLine demo, 74–75
 - DirectX, performance, 148
 - Dirty regions, 251
 - Discrete interpolation, 367–368, 409–411
 - DiscreteInterpolator class, 367–368
 - Displays
 - maximum refresh rate, 289
 - prototyping. *See* Screen mockups.
 - dispose(), 174–175
 - Double-buffering, 28–31, 334
 - Drag-and-drop, 238, 462
 - DRAG_LAYER, 238
 - draw(), 78
 - drawImage(), 76–77, 102–103
 - Drawing
 - fill *versus* draw, 87–89
 - rectangles, 78
 - shapes, 80–83
 - drawLine(), 77
 - drawRect(), 78
 - DrawShapes demo, 80–83
 - drawString(), 78
 - Drop shadow
 - AlphaComposite, 167
 - description, 437–438
 - realistic, 440–442
 - simple, 438–440
 - Dst rule, 157
 - DstAtop rule, 158
 - DstIn rule, 158
 - DstOut rule, 159
 - DstOver rule, 159
 - Duff, Tom, 153
 - Duration of animation, 345, 350–351
 - Dynamic effects. *See also* Static effects.
 - bloom, 6, 478–482
 - fading. *See also* Transitions.
 - AlphaComposite class, 468–470
 - color, 470–472
 - cross, 472
 - demo, 468–470
 - description, 465–467
 - library of components for, 472
 - strategies for, 467–468
 - morphing
 - buttons, 491–495
 - demo, 491–495
 - description, 489–491
 - shape tweening, 490
 - motion
 - description, 460–465
 - drag and drop, 462
 - moving objects, 462–465
 - undo/redo syndrome, 460
 - pulsating
 - bloom, 6, 478–482. *See also* Glow.
 - demos, 474–478, 482–484
 - description, 473–474
 - glow, 474–478. *See also* Bloom.
 - spring, 484–489
- ## E
- EDT (Event Dispatch Thread). *See also* Events.
 - blocking, 31–32
 - deadlock, 35–37
 - definition, 31
 - description, 16
 - identifying, 35
 - perceived performance, 31–32
 - posting a runnable task, 34, 35–37
 - thread safety, 33–37
 - Timer class, 37–38
 - timers, 37–38
 - Effects
 - animated. *See* Dynamic effects.
 - CompositeEffect, 512–513
 - custom, creating, 513–516
 - description, 509
 - Effects API, 510–516
 - FadeIn, 510
 - FadeOut, 510
 - init(), 513
 - Move, 510, 514
 - nonanimated. *See* Static effects.

paint(), 513
 removing, 512
 Rotate, 510, 515–516
 setEffect(), 511–513
 setup(), 513
 transition types, 511–513
 Unchanging, 511
 using, 511–513
 Effects API, 510–516
 end(), 348–349
 End caps, 51
 Evaluator class, 399–401
 Event Dispatch Thread (EDT). *See* EDT
 (Event Dispatch Thread).
 Event handling. *See* Triggers.
 Events. *See also* EDT (Event Dispatch
 Thread).
 automation timing, 348–350, 495
 demos, 311–312
 glass pane, 231–235
 input, blocking, 230–235
 mouse, 231–235
 rendering, 16
 Examples. *See* Demos.

F

FadeIn effect, 510
 FadeOut effect, 510
 Fading. *See also* Transitions.
 AlphaComposite class, 468–470
 buttons, demo, 353–356
 color, 470–472
 cross, 472
 demo, 468–470
 description, 465–467
 in/out, 186
 library of components for, 472
 strategies for, 467–468
 FadingButton demo, 353–356
 Farin, Gerald E., 377
 Feiner, Steven, 377
 fill(), 78
 FillDraw demo, 87–89

Filling
 demo, 87–89
 versus drawing, 87
 fillRect(), 78
 Filters
 affine transformations, 203–204
 AffineTransformOp, 203–204
 applying to multiple images, 202
 black border, removing, 209–211
 brightening images, 213–214
 BufferedImageOp, 201–203, 214–221
 color model conversion, 204–206,
 211–213
 ColorConvertOp, 204–206
 ConvolveOp, 206–211
 convolving images, 206–211
 creating, 214–221
 darkening images, 213–214
 edge problems, 209–211
 grayscale conversion, 204–206
 immutability, 217
 LookupOp, 211–213
 mapping source/destination color values,
 211–213
 offscreen processing, 201–202
 overview, 200
 at painting time, 201
 performance, 219, 220, 222
 picture shape, 203–204
 RescaleOp, 213–214, 445–446
 scaling color values, 213–214
 scaling pictures, 203–204
 tinting images, 213–214, 214–221
 Filthy Rich Clients
 definition, 1, 3–7
 effectives, 3–7
 overview, 3–7
Filthy Rich Clients, online resources,
 551
 Firing triggers, 381
 Fixed delay, 285
 Fixed rate, 285
 Floating-toolbars layer, 238
 FocusTrigger, 385–386

Foley, James D., 377
 FontHints demo, 59–68
 Fonts, graphics state, 49, 50–51
 Foreground color, graphics state, 49, 50
 Frame rates

- frame-based animation, 268
- minimum required, 289–290
- smoothing, 318, 332

 Frame-based animation. *See* Animation, frame-based.

G

Gaussian blurs, 428–432
The Geometry Toolbox for Graphics and Modeling, 377
 getBackground(), 50
 getColor(), 50
 getComposite(), 62–63
 getFasterScaledInstance(), 111–113
 getFont(), 50–51
 getGraphics(), 47–48
 getHSBColor(), 444–445
 getInterval(), 414
 getPaint(), 63
 getRenderingHint(), 53
 getRGB(), 219
 getScaledInstance()

- performance, 102, 107, 109
- “The Perils of Image.getScaledInstance()”, 98

 getStroke(), 51
 getSubimage(), 120
 Glass pane. *See also* Layered panes.

- blocking input events, 230–235
- in the JFrame hierarchy, 224
- limitations, 237
- mouse events, 231–235
- optimized painting, 227–230
- overview, 223–225
- painting on, 225–230
- root panes, 223
- transitions, 527–528
- visibility, 225

 Glow, 474–478
 GradientPaint class, 187–188
 Gradients. *See also* Highlights; Reflection.

- acyclic, 195–197
- bright spots, 189–192
- caching, 193–195
- circular, 189–192
- cyclic, 195–197
- dual (multistops linear), 187–189
- fading in/out, 186
- nonhorizontal, 195–197
- nonvertical, 195–197
- optimizing, 193–197
- performance, 193–197
- radial, 189–192
- reflection, 182–186
- regular (two-stops linear), 179–186
- screen mockups, 547
- shiny surfaces, 187–189
- simulated depth, 180–181
- specular highlights, 189–192
- spheres, 189–192

 Graham, Jim, 111, 492
 Graphics. *See also* Images.

- antialiasing
 - demo, 56
 - description, 54–56
 - jaggies, 54
 - shapes, 57
 - text, 57–59
- borders, on components, 75
- copying areas, 83
- demos
 - color, graphics primitives, 63–64
 - copying areas, 83–87
 - custom components, matching desktop settings, 59–68
 - diagonal line, 74–75
 - donut shape, 79
 - drawing, 87–89
 - drawing shapes, 80–83
 - filling, 87–89
 - image scaling, 63–64, 67–68
 - rotating objects, 63–64, 67–68, 68–74
 - text hints, setting, 59–68

- transformations, 63–64, 67–68
 - translating objects, 63–64, 67–68
 - drawing, 87
 - filling, 87
 - matrix math, 64–67
 - rectangles, drawing and filling, 78
 - rendering. *See also* Rendering, with Swing.
 - description, 45–46
 - getting the Graphics object, 46–48
 - into images, 47–48
 - temporary state change, 46–47
 - rendering hints
 - definition, 49
 - image scaling, 53–54
 - KEY_ANTIALIASING, 57
 - KEY_INTERPOLATION, 53–54
 - KEY_TEXT_ANTIALIASING, 57–59
 - shape antialiasing, 57
 - text antialiasing, 57–59, 59–68
 - VALUE_ANTIALIAS, 57
 - VALUE_INTERPOLATION, 53–54
 - VALUE_TEXT_ANTIALIAS, 58–59
 - shapes, drawing and filling, 78
 - Graphics object
 - changing, 70
 - getting, 46–48
 - versus* Graphics2D object, 44–45
 - Graphics state
 - background color, 49, 50
 - clips, 49, 61–62
 - clobbered, restoring, 73
 - clobbering, 72
 - composite, 49
 - description, 48–49
 - fonts, 49, 50–51
 - foreground color, 49, 50
 - image scaling, 53–54
 - paint property, 49
 - persistence, 71–72
 - properties, 48–51
 - setting, 51–54
 - stroke, 49, 51
 - temporary change, 46–47
 - transform property, 49
 - Graphics2D object, 44–45
 - Grayscale conversion, 204–206
 - GUI toolkits, 12–13
 - GUIs, animation, 303–304
- ## H
- Hansford, Dianne, 377
 - Hard edges, animation, 326–327
 - High resolution pictures, scaling, 109–111
 - Highlights (emphasis), on text, 448–450
 - Highlights (lighting). *See also* Gradients.
 - brightening, 444–446
 - description, 442–444
 - images, 445–446
 - text, 444–445
 - Hints
 - image scaling, 102–103
 - rendering
 - definition, 49
 - image scaling, 53–54
 - KEY_ANTIALIASING, 57
 - KEY_INTERPOLATION, 53–54
 - KEY_TEXT_ANTIALIASING, 57–59
 - shape antialiasing, 57
 - text antialiasing, 57–59, 59–68
 - VALUE_ANTIALIAS, 57
 - VALUE_INTERPOLATION, 53–54
 - VALUE_TEXT_ANTIALIAS, 58–59
 - Honoring the clip, 117–120
 - HSB color, converting to RGB, 444–445
 - Hughes, John F., 377
- ## I
- Image files, 94
 - Image I/O, 94
 - Image objects, 92
 - ImageHolder object, 523–525
 - Images. *See also* Graphics.
 - 3D, 134–142
 - brightening, 213–214, 445–446
 - BufferedImage objects
 - advantages of, 95–96
 - converting images to, 96–97

- Images, `BufferedImage` objects, *continued*
 - definition, 93
 - transparent/translucent images, 97
 - the clip, 115–121
 - clipping
 - antialiased-clipping, 168
 - the clip, 115–121
 - soft-clipping, 168
 - compatible
 - converting to, 124–126
 - definition, 94
 - performance, 121–126
 - uses for, 122–123
 - convolving, 206–211
 - darkening, 213–214
 - definition, 91
 - drawing, 76–77
 - filters. *See* Filters.
 - image files, 94
 - image I/O, 94
 - Image objects, 92
 - intermediate
 - caching, 141
 - definition, 94
 - image-based rendering, 134
 - performance, 134–142
 - translucency, 142
 - loading images from a hard drive, 39–41
 - managed
 - advantages of, 128–129
 - definition, 94
 - frequent rendering, 132–133
 - grabbing the `DataBuffer`, 129–132
 - history of, 126
 - performance, 126–133
 - mipmapping, 524
 - surface loss, 93
 - tinting, 213–214, 214–221
 - toolkit, 93–94
 - types of, 92–95
 - `olatileImage` objects, 92–93
- Images, scaling
 - affine transforms, 100
 - area averaging, 103–104
 - bicubic, 54, 102–104
 - bilinear, 54, 102–104
 - demos
 - high resolution pictures, 109–111
 - progressively larger images, 105–109
 - `drawImage()`, 77, 102–103
 - `getFasterScaledInstance()`, 111–113
 - `getScaledInstance()`, 102, 107, 109
 - graphics state, 53–54
 - hints for, 102–103
 - matrix math, 64–67
 - nearest neighbor, 53–54, 102–104
 - to a new image, 101
 - overview, 98–99
 - prescaled, 524
 - progressive bilinear, 104
 - quality *versus* performance, 101–104
 - rendering hints, 53–54
 - subregions of images, 100
 - translation, 100
- Immutability, 173, 217
- `init()`, 513
- Interframe volatility, 277–278
- Intermediate images
 - caching, 141
 - definition, 94
 - image-based rendering, 134
 - performance, 134–142
 - translucency, 142
- IntermediateImages demo, 137–142
- `interpolate()`, 366–367
- Interpolation
 - acceleration, 360–363
 - anchor points, 369
 - control points, 369
 - deceleration, 360–363
 - definition, 345
 - demo, 363–364
 - description, 359–360
 - `setAcceleration()`, 360–363
 - `setDeceleration()`, 360–363

Interpolator class
 custom interpolation, 377–378
 demos, 372–378
 description, 364–365
 discrete interpolation, 367–368
DiscreteInterpolator class, 367–368
 interpolate(), 366–367
 key frames, 408–411
 linear interpolation, 366–367
LinearInterpolator class, 366–367
 spline interpolation, 368–372
SplineInterpolator class, 368–372
 Timing Framework, 408–411
 types of, 365–372
 uses for, 365
Interpolator race demo, 363–364
 invokeAndWait(), 35–37
 invokeLater(), 34
 isEventDispatchThread(), 35
 isRunning(), 347

J

Jaggies, 54
Java 2D
 description, 13
 graphics capabilities, 43–45
 Graphics object, 44–45
 Graphics2D object, 44–45
Java sites, 550–551
JLayeredPane component. *See also* Layered panes.
 alternative to, 244–247
 bottommost layer, 238
 buttons layer, 238
 combo-box drop-down layer, 238
 drag-and-drop operations, 238
 DRAG_LAYER, 238
 floating-toolbars layer, 238
 layers, 238
 layout manager, 243–244
 menus layer, 238
 modal dialog layer, 238
 MODAL_LAYER, 238

ordering components, 242–243
 overview, 238–242
 PALETTE_LAYER, 238
 palettes layer, 238
 popup window layer, 238
 relative values, 241–242
 tables layer, 238
 tooltips layer, 238
 warning messages, 238–239
JOGL project, online resources, 550

K

Key frames, 408–411
KEY_ANTIALIASING hint, 57
Keyboard animation commands, 338–341
KeyFrames class
 constructors, 411–414
 demo, 414–419
 description, 402–404
 getInterval(), 414
 helper classes, 404–411
 Interpolator class, 408–411
 KeyTimes class, 407–408
 KeyValues class, 404–407
 methods, 414
KEY_INTERPOLATION hint, 53–54
KEY_TEXT_ANTIALIASING hint, 57–59
KeyTimes class, 407–408
KeyValues class, 404–407
Knuth, Donald, 197

L

Layered panes. *See also* Glass pane;
 JLayeredPane component.
 bottommost layer, 238
 buttons layer, 238
 combo-box drop-down layer, 238
 drag-and-drop operations, 238
 floating-toolbars layer, 238
versus glass pane, 237
 layers, 238
 layout manager, 243–244

- Layered panes, *continued*
 - menus layer, 238
 - modal dialog layer, 238
 - ordering components, 242–243
 - overview, 238–242
 - palettes layer, 238
 - popup window layer, 238
 - relative values, 241–242
 - tables layer, 238
 - tooltips layer, 238
 - warning messages, 238–239
 - Layout manager, 243–244, 507, 522
 - Libraries
 - Animated Transitions library, 497–530, 550
 - Timing Framework, 343–420, 549
 - Lighting highlights. *See* Highlights (lighting).
 - Linear interpolation, 366–367
 - Linear shapes, animation, 332
 - LinearGradientPaint class, 188–189
 - LinearInterpolator class, 366–367
 - Lines
 - attributes, 51
 - end caps, 51
 - width, 51
 - width, and performance, 51
 - Listeners, adding, 380–381
 - LookupOp filter, 211–213
- M**
- Main thread, 31–32
 - Managed images
 - advantages of, 128–129
 - definition, 94
 - frequent rendering, 132–133
 - grabbing the DataBuffer, 129–132
 - history of, 126
 - performance, 126–133
 - Matrix math, 64–67
 - Menus layer, 238
 - Milliseconds, 276–278, 291–293
 - Mipmapping, 524
 - Mockups. *See* Aerith.
 - Modal dialog layer, 238
 - MODAL_LAYER, 238
 - Monitors. *See* Displays.
 - Morphing
 - buttons, 491–495
 - demo, 491–495
 - description, 489–491
 - shape tweening, 490
 - Morphing2D class, 492–495
 - Motion blur, 328–329
 - Motion effects. *See also* Animation.
 - description, 460–465
 - drag and drop, 462
 - moving objects, 462–465
 - undo/redo syndrome, 460
 - MouseTrigger, 388–390
 - Move effect, 510, 514
- N**
- Nanoseconds, 278, 291–293
 - nanoTime(), 278, 291–293
 - Nearest neighbor image scaling, 102–104
 - Nonhorizontal gradients, 195–197
 - Nonvertical gradients, 195–197
- O**
- Online demos. *See* Demos.
 - Online resources. *See also* Books and publications.
 - Adobe kuler, 547
 - Aerith project, 550
 - Animated Transitions library, 550
 - authors' sites, xxiii, 551
 - Color Schemer Online, 546
 - Color Schemer Studio, 546
 - colors, 546–547
 - COLOURlovers, 547
 - Filthy Rich Clients*, 551
 - Java 2D FAQ, 145
 - Java sites, 550–551
 - JOGL project, 550
 - SMIL (Synchronized Multimedia Integration Language), 377

- Swing component library, 472
 - SwingLabs project, 550
 - Timing Framework library, 549
 - Opacity. *See also* Translucency; Transparency; Visibility.
 - drop shadow, 167–168
 - Java 2D, 27–28
 - painting, 21, 27–28
 - rendering, 27–28
 - OpenGL pipeline, 147–148
 - Optimizing
 - animation resolution, 318–319
 - gradients, 193–197
 - painting. *See* RepaintManager.
 - painting glass pane, 227–230
 - Ordering components. *See* JLayeredPane component; Layered panes.
- P**
- paint()
 - calling directly, 48
 - custom effects, 513
 - description, 19–20
 - overriding, 21, 24–27
 - Paint property, graphics state, 49
 - paintComponent(), 21–24
 - Painter’s algorithm, 21
 - Painting. *See also* paint(); repaint().
 - asynchronous requests, 17–19
 - coalescing requests, 18
 - description, 17
 - entire components, 18
 - on the glass pane, 225–230
 - opacity, 21, 27–28
 - optimizing. *See* RepaintManager.
 - over the user interface. *See* Glass pane.
 - setOpaque(), 21, 27–28
 - subregions of components, 18
 - synchronous requests, 19–20
 - PALETTE_LAYER, 238
 - Palettes layer, 238
 - Papers. *See* Books and publications.
 - pause(), 347
 - Pausing an animation, 347
 - Perceived performance
 - animation, 316
 - EDT (Event Dispatch Thread), 31–32
 - threading, 31–32
 - Performance. *See also* Perceived performance.
 - animation, 317–318
 - blur, 433–434
 - the clip, 115–121
 - command-line flags, 145–149
 - compatible images, 121–126
 - copying areas, 83–87
 - debugging, 148–149
 - demos, 137–142
 - diagonal lines, 120
 - DirectX, 148
 - EDT (Event Dispatch Thread), 31–32
 - filters, 219, 220, 222
 - gradients, 193–197
 - honoring the clip, 117–120
 - intermediate images, 134–142
 - line width, 51
 - managed images, 126–133
 - OpenGL pipeline, 147–148
 - versus* quality, image scaling, 101–104
 - rendering, 143–145, 146–148
 - repaint(), 227
 - threading, 31–32
 - timer resolution, 288, 290, 299–300
 - transitions, 529–530
 - “The Perils of Image.getScaledInstance(),” 98
 - Petrucci, Sébastien, 441
 - Photographs
 - browsing, 519–526
 - scaling, 105–109
 - Picture frames, 167
 - PictureScaler demo, 109–111
 - Pixel color change, 322–323
 - Pixels, colors, 211
 - Popup window layer, 238
 - Porter, Thomas, 153
 - Porter-Duff equations, 153–154
 - Porter-Duff rules, 155–162

Posting a runnable task, 34, 35–37
 Prescaled images, 524
 Progressive bilinear image scaling, 104
 Properties, graphics state, 48–51
 Property setters
 definition, 392
 demos, 393–395
 getter, 394
 known types, 395
 object access, 394
 setter, 394
 PropertySetter class
 constructors, 395–397
 definition, 395
 demo, 398
 evaluating intermediate types,
 399–401
 Evaluator class, 399–401
 factory methods, 397–398
 TimingTarget methods, 398
 Prototyping. *See* Aerith.
 Publications. *See* Books and publications.
 Pulsating effects
 bloom, 6, 478–482
 demos, 474–478, 482–484
 description, 473–474
 glow, 474–478

Q

Querying running animations, 347

R

Racetrack demo, 356–359
 Radial gradients, 189–192
 RadialGradientPaint class, 189–192
 rasterStolen variable, 132
 Realistic drop shadow, 440–442
 Realistic motion, 319–320
 Reflection. *See also* Gradients.
 blurred, 435–437
 demos, 253–262
 description, 434
 drawing, 435
 simulating, 182–186
 Regular (two-stops linear) gradients,
 179–186
 Rendering
 animation, 337–341
 description, 45–46
 getting the Graphics object, 46–48
 image-based, 134
 into images, 47–48
 managed images, 132–133
 performance, 143–145, 146–148
 temporary state change, 46–47
 text in drawing areas, 78
 Rendering, with Swing. *See also*
 Threading.
 back buffers, 29
 customized, 21–24
 demos
 buttons, painting, 23–24
 buttons, translucent, 25–27
 filling a blank canvas, 22–23
 description, 20–21
 double-buffering, 28–31
 events, 16
 flicker, 29–30
 opacity, 27–28
 paintComponent(), 21–24
 painter's algorithm, 21
 painting. *See also* paint(); repaint().
 asynchronous requests, 17–19
 coalescing requests, 18
 description, 17
 entire components, 18
 opacity, 21, 27–28
 setOpaque(), 21, 27–28
 subregions of components, 18
 synchronous requests, 19–20
 translucency, 27–28
 triple-buffering, 28
 Rendering hints
 definition, 49
 image scaling, 53–54
 KEY_ANTIALIASING, 57

- KEY_INTERPOLATION, 53–54
 - KEY_TEXT_ANTIALIASING, 57–59
 - shape antialiasing, 57
 - text antialiasing, 57–59, 59–68
 - VALUE_ANTIALIAS, 57
 - VALUE_INTERPOLATION, 53–54
 - VALUE_TEXT_ANTIALIAS, 58–59
 - repaint(), 18–19, 227
 - RepaintManager, 251–262
 - repeat(), 349
 - Repeating animations
 - definition, 345
 - description, 351–352
 - setRepeatBehavior(), 351–352
 - setRepeatCount(), 351–352
 - Replacing existing drawings, 161, 166–168
 - RescaleOp filter, 213–214, 445–446
 - Resolution
 - animation, optimizing, 318–319
 - definition, 345
 - description, 352
 - setResolution(), 352
 - smoothing, 318–319
 - timer. *See* Timer resolution.
 - Timing Framework, 345, 352
 - resume(), 347
 - Resuming an animation, 347
 - RGB color, converting to HSB, 444–445
 - RGBtoHSB(), 444–445
 - Rich clients, definition, 2
 - Root panes, 223
 - rotate(), 63–64, 67–68
 - Rotate effect, 510, 515–516
 - Rotating objects, 63–64, 67–68, 68–74
 - RotationAboutCenter demo, 68–74
 - Rules for colors, transparency, translucency.
 - See* AlphaComposite;
 - Composites.
- S**
- scale(), 63–64, 67–68
 - ScaleTest demo, 105–109
 - Scaling
 - color values, 213–214
 - images. *See* Images, scaling.
 - pictures, 203–204
 - Screen mockups. *See also* Aerith.
 - coding from, 540–544
 - colors, 545–547
 - gradients, 547
 - guides, 543
 - required skills, 544–545
 - workflow design, 538–539
 - Screens. *See* Displays.
 - ScreenTransition object, 525–526
 - setAcceleration(), 360–363
 - setBackground(), 50
 - setClip(), 61
 - setColor(), 50
 - setComposite(), 62–63
 - setDeceleration(), 360–363
 - setDuration(), 350–351
 - setEffect(), 511–513
 - setFont(), 50–51
 - setOpaque(), 21, 27–28
 - setPaint(), 63
 - setRenderingHint(), 53
 - setRepeatBehavior(), 351–352
 - setRepeatCount(), 351–352
 - setResolution(), 352
 - setStartDelay(), 352
 - setStartDirection(), 352
 - setStartFraction(), 353
 - setStroke(), 51
 - setTransform(), 63–64
 - setup(), 513
 - setupNextScreen(), 527–528
 - Shapes
 - antialiasing, 57
 - drawing, 78
 - filling, 78
 - tweening, 490. *See also* Morphing.
 - Sharpening
 - description, 450–451
 - downscaled images, 455–458

- Sharpening, *continued*
 - simple, 452–454
 - USM (unsharp masking), 454
- Sharpening kernels, 454
- Shiny surfaces, simulating, 187–189
- `sleep()`, 279–280, 293–297
- Sleeping, 279–282, 293–297
- SMIL (Synchronized Multimedia Integration Language), 377
- Smoothing edges. *See* Antialiasing.
- SmoothMoves demo, 335–341
- Soft-clipping, 168
- Source pixels, 156
- Specular highlights, simulating, 189–192
- Spheres, simulating, 189–192
- Spline interpolation, 368–375, 376–378
- `SplineEditor` demo, 376–378
- `SplineInterpolator` class, 368–372
- `SplineInterpolatorTest` demo, 372–375
- Spotlighting, 446–448
- Spring effect, 484–489
- Src rule, 160
- SrcAtop rule, 160
- SrcIn rule, 161, 166–168
- SrcOut rule, 161
- SrcOver rule, 162, 165–166
- `start()`, 347
- Start delay, 352
- Start direction, 352
- Start fraction, 353
- Starting animations
 - definition, 345
 - `setStartDelay()`, 352
 - `setStartDirection()`, 352
 - `setStartFraction()`, 353
 - `start()`, 347
 - start delay, 352
 - start direction, 352
 - start fraction, 353
- Static effects. *See also* Dynamic effects.
 - blur
 - bokeh, 423–424
 - box, 426–428
 - depth of field, 423
 - description, 423–425
 - Gaussian, 428–432
 - performance, 433–434
 - simple, 426–428
 - drop shadow
 - description, 437–438
 - realistic, 440–442
 - simple, 438–440
 - highlights (emphasis), on text, 448–450
 - highlights (lighting)
 - brightening, 444–446
 - description, 442–444
 - images, 445–446
 - text, 444–445
 - reflection
 - blurred, 435–437
 - description, 434
 - drawing, 435
 - sharpening
 - description, 450–451
 - downscaled images, 455–458
 - simple, 452–454
 - USM (unsharp masking), 454
 - sharpening kernels, 454
 - spotlighting, 446–448
- `stop()`, 347
- Stopping animations, 347, 353
- Straight edges, animation, 327–328
- Stroke, graphics state, 49, 51
- Surface loss, 93
- Swing, 13. *See also* Rendering, with Swing.
- SwingLabs project, online resources, 550
- `SwingUtilities` class, 38–41
- `SwingWorker` class, 38–41
- Synchronized Multimedia Integration Language (SMIL), 377
- Synchronous paint requests, 19–20
- System Properties for Java 2D...*, 145

T

- Tables layer, 238
- Tearing artifact, 330
- Temporary offscreen images, 170
- Text
 - antialiasing, 57–59, 59–68
 - brightening, 444–445
 - in drawing areas, 77
 - highlights (emphasis), 448–450, 482–484
 - highlights (lighting), 444–445
 - hints, setting, 59–68
 - pulsating border, 482–484
- Thin clients, 1
- Thread safety, 33–37
- Threading. *See also* Rendering.
 - deadlocks, 35–37
 - description, 31–32
 - EDT (Event Dispatch Thread)
 - blocking, 31–32
 - deadlock, 35–37
 - definition, 31
 - description, 16
 - identifying, 35
 - perceived performance, 31–32
 - posting a runnable task, 34, 35–37
 - thread safety, 33–37
 - Timer class, 37–38
 - timers, 37–38
 - `invokeAndWait()`, 35–37
 - `invokeLater()`, 34
 - `isEventDispatchThread()`, 35
 - loaded file names, displaying, 39–41
 - loading images from a hard drive, 39–41
 - main thread, 31–32
 - perceived performance, 31–32
 - `SwingUtilities` class, 38–41
 - `SwingWorker` class, 38–41
 - toolkit thread, 31–32
 - utility classes for, 38–41
- Threading model, 33–37
- Time-based animation. *See* Animation,
 - time-based.
- Timer (`java.util`), 37–38, 282
- Timer (`javax.swing`), 38, 284–288
- Timer resolution
 - `currentTimeMillis()`, 291–293
 - definition, 288–289
 - increasing, 278
 - measuring, 297–299
 - milliseconds, 276–278, 291–293
 - nanoseconds, 278, 291–293
 - `nanoTime()`, 291–293
 - performance, 288, 290, 299–300
 - sleeping, 293–297
- Timer utilities
 - callbacks, 284–285
 - `currentTimeMillis()`, 276–278
 - demos, 282–284, 286–288
 - fixed delay, 285
 - fixed rate, 285
 - interframe volatility, 277–278
 - milliseconds, 276–278
 - nanoseconds, 278
 - `nanoTime()`, 278
 - querying current time, 275–278
 - `sleep()`, 279–280, 293–297
 - sleeping, 279–282, 293–297
 - Timer (`java.util`), 282–284
 - Timer (`javax.swing`), 284–288
 - timers, 280–284
 - `wait()`, 279–280, 295–297
 - wake-up calls, 279–282
- Timers, 37–38, 280–284
- Timing effects, animation, 317–320,
 - 335–337
- Timing events, receiving, 348–350
- Timing Framework
 - `Animator` class, 345–347
 - autoboxing, 405
 - callbacks. *See also* `TimingTarget` object.
 - cleanup operations, 348–349
 - definition, 345
 - timing events, receiving, 348–350
 - timing targets, 349–350
 - controlling running animations, 347–348
 - definition, 344

Timing Framework, *continued*

demos

- bouncing spheres, 382–392
- fading button, 353–356
- interpolator race, 363–364
- KeyFrames class, 414–419
- multistep race, 414–419
- property setters, 393–395
- racetrack, 356–359, 363–364, 391–392, 393–395, 398–399, 414–419
- spline editor, 376
- trigger race, 391–392
- triggers, 382–392

duration of animation, 345, 350–351

ending behavior, 345

interpolation

- acceleration, 360–363
- anchor points, 369
- control points, 369
- deceleration, 360–363
- definition, 345
- demo, 363–364
- description, 359–360
- setAcceleration(), 360–363
- setDeceleration(), 360–363

Interpolator

- custom interpolation, 377–378
- demos, 372–378, 414–419
- description, 364–365
- discrete interpolation, 367–368
- DiscreteInterpolator class, 367–368
- interpolate(), 366–367
- linear interpolation, 366–367
- LinearInterpolator class, 366–367
- spline interpolation, 368–372
- SplineInterpolator class, 368–372
- types of, 365–372
- uses for, 365

key frames, 402–414

KeyFrames class

- constructors, 411–414
- demo, 414–419

description, 402–404

- getInterval(), 414
- helper classes, 404–411
- Interpolator class, 408–411
- KeyTimes class, 407–408
- KeyValues class, 404–407
- methods, 414

online resources, 549

pausing an animation, 347

property setters

- definition, 392
- demos, 393–395
- getter, 394
- known types, 395
- object access, 394
- setter, 394

PropertySetter class

- constructors, 395–397
- definition, 395
- demo, 398
- evaluating intermediate types, 399–401
- Evaluator class, 399–401
- factory methods, 397–398
- TimingTarget methods, 398

querying running animations, 347

repetition of animations, 345, 351–352

resolution, 345, 352

resuming an animation, 347

setDuration(), 350–351

starting an animation, 347

starting behavior, 345, 352–353

stopping an animation, 347

TimingTarget object, 348–350. *See also* Callbacks.

triggers

- ActionTrigger, 384
- auto-reversing, 381
- built-in, 382–384
- canceling, 381
- creating, 380
- definition, 379–380
- demos, 382–392

- firing, 381
- FocusTrigger, 385–386
- listeners, adding, 380–381
- MouseEvent, 388–390
- superclasses, 381–382
- TimingTrigger, 389–390
- Trigger class, 381–382
- TriggerEvent class, 382
- uses for, 344
- Timing targets, 349–350
- timingEvent(), 349
- TimingTarget methods, 398
- TimingTarget object, 348–350
- TimingTargetAdapter class, 350
- TimingTrigger, 389–390
- Tinting images, 213–214, 214–221
- Toolkit images, 93–94
- Toolkit thread, 31–32
- Toolkits for GUIs. *See* GUI toolkits.
- Tooltips layer, 238
- transform(), 63–64
- Transform property, graphics state, 49
- Transformations, 63–68
- Transitions. *See also* Animated transitions; Fading.
 - Animated Transitions library, 501
 - AnimatedTransitions API, 502–503
 - between application states, 497–501
 - effects
 - CompositeEffect, 512–513
 - custom, creating, 513–516
 - description, 509
 - Effects API, 510–516
 - FadeIn, 510
 - FadeOut, 510
 - init(), 513
 - Move, 510, 514
 - paint(), 513
 - removing, 512
 - Rotate, 510, 515–516
 - setEffect(), 511–513
 - setup(), 513
 - transition types, 511–513
 - Unchanging, 511
 - using, 511–513
 - glass pane, 527–528
 - GUI states, 501–502
 - image browser, 519–522
 - layout changes, animating, 528–529
 - layout details, hiding, 527–528
 - next-screen setup, 527–528
 - performance, 529–530
 - search engine, 503–509, 516–518
 - setupNextScreen(), 527–528
- translate(), 63–64, 67–68
- Translating objects, 63–64, 67–68
- Translation, image scaling, 100
- Translucency. *See also* Opacity; Transparency.
 - BufferedImage objects, 97
 - demo, 249–251
 - effect, 162, 165–166
 - intermediate images, 142
 - painting, 27–28
 - rules for. *See* AlphaComposite; Composites.
- TranslucentPanel demo, 249–251
- Transparency, 97. *See also* Opacity; Translucency.
- Trigger class, 381–382
- TriggerEvent class, 382
- Triggers
 - ActionTrigger, 384
 - auto-reversing, 381
 - built-in, 382–384
 - canceling, 381
 - creating, 380
 - definition, 379–380
 - demos, 382–392
 - firing, 381
 - FocusTrigger, 385–386
 - listeners, adding, 380–381
 - MouseEvent, 388–390
 - superclasses, 381–382
 - TimingTrigger, 389–390
 - Trigger class, 381–382
 - TriggerEvent class, 382

Triple-buffering, 28
 Tweening. *See* Morphing.
 Two-stops linear (regular) gradients,
 179–186

U

Unchanging effect, 511
 Undo/redo syndrome, 460
 UnsharpMaskFilter class, 455–458
 User interface design, books and
 publications, 547
 USM (unsharp masking), 454

V

VALUE_ANTI_ALIAS hint, 57
 VALUE_INTERPOLATION hint, 53–54
 VALUE_TEXT_ANTI_ALIAS hint, 58–59
 Van Dam, Andries, 377
 VDTs. *See* Displays.
 Vector drawings, scaling, 105–109
 Vertical retrace effect, 329–334
 Violet, Scott, 18, 46

Visibility. *See also* Opacity; Translucency;
 Transparency.
 the clip, 115–121
 glass pane, 225
 Vision feature, 535–537
 Visual design, books and publications,
 547
 VolatileImage objects, 92–93

W

wait(), 279–280, 295–297
 Wake-up calls, 279–282
 Warning messages, 238–239
 Web clients, 2
 Web resources. *See* Online resources.
 Workflow design. *See also* Aerialith.
 with a graphics editor, 535–537
 on paper, 533–535, 537–538
 screen mockups, 538–543

X

Xor rule, 162