

# Index

## Numbers

**1st person camera perspective, 323–324**

**2.5D, 6**

**2D**

billboarding surfaces to look 3D, 242–244  
 converting website to 3D. See CSIS (Center for Strategic and International Studies) website rendering to 3D, 8–9

**2nd person camera perspective, 323–324**

**3D. See also Flash 3D, understanding**

adding to Flickr menu. See Flickr menu  
 converting 2D to. See 2D  
 creating Photoshop layer in, 511  
 Flash 10 engine in 13 lines of code, 14  
 Flash 9 engine. See Flash 9, building 3D engine in 18 lines of code  
 models. See models, 3D  
 physics. See physics, 3D  
 rotation. See rotation, 3D  
 Wall Carousel Course. See Wall Carousel Course, 3D  
 wall course. See Wall Carousel Course, 3D  
 websites. See Websites, developing 3D

**3DSMax**

billboarding surfaces to look 3D, 242–244  
 building ray gun in, 377–378  
 converting file into MD2 format, 185–186  
 creating model in, 178, 195  
 creating multiple images on single object, 191  
 creating single and multiple objects with simple images, 188–191  
 primitives, 75  
 simple exporters, 204

**3rd person camera perspective, 323–324**

## A

**acceleration**

adding to `DisplayObject3D` class, 326–328  
 calculating for camera, 322–323  
 slot car racing game, adding FMS, 444–456

slot car racing game, having a wreck, 442–444  
 Wiimote accelerometer sensing, 399–400

**access modifiers (scope), 48–49**

**Action Script 3.0 Game Programming University (Rozenzweig), 238**

**action tokens, 282–283**

**ActionScript**

code for particle systems, 230–231  
 running Papervision3D in Flex as, 41–43  
 simulating 3D objects in, 34  
 turning Flash animation into, 334  
 turning timeline animation into, 29–31

**addChild method**

adding models to stage, 184  
 adding sphere to scene, 57  
 adding sphere to stage, 54  
 in final base code, 55–56  
 formulating PV3D class, 52  
 setting viewport, 53  
 using `rawChildren` tag with, 43

**addEventListener method, 17–19**

coding animation, 11–12  
 creating cube in CS4, 116  
 creating parametric particle system, 17–19

**addPropertyArray method, 31**

**addWorldForce method, 390**

**Adobe**

AIR application. See AIR  
 Flash Collaboration Service, 457–458  
 Flex application. See Flex  
 Photoshop. See Photoshop

**affine transformations, 134–135**

**AI (Artificial Intelligence), adding to Pong game, 408**

**AIR, 285–317**

building modeling program. See modeling program, PV3D  
 fixing Flash Sandbox security issue, 255–259  
 Google map wormhole created as, 608–609  
 overview of, 285  
 preventing insecurity of swf files, 422  
 sculpting prim with `face3d`, 313–316

## **AIR, building editor in, 306–312**

- accessing local file system, 307
- deployment, 312
- overview of, 306–307
- saving bitmap data, 307–310
- saving to local PC using Flash 10, 310–311
- turning Webcam into PV3D heightmap, 311–312
- using SQLite, 312

## **Alchemy**

- difficulty optimizing PV3D for Flash 10, 591
- Flash importing C/C++ libraries using, 251
- information on, 201

## **Alert box, saving bitmap data, 308**

## **TheAlgorithmist, 438**

## **alignment, stage, 65–66**

## **alpha property, ParticleMaterial, 222–223**

## **ambientColor variable, CS4 flat shading, 595**

## **Animate Super class, Flash Builder, 506**

## **AnimateTransform class, 517–518**

## **animation, 3D**

- Collada parsers and, 198–200
- of exploding balls, 382–383
- inside Collada files, 190
- of light source in Pixel Bender, 218
- loop, 140–141
- in MD2 and Collada, 186–187
- of muzzles, 378–379
- principles of, 28–29
- recording. *See* recording 3D animations
- of skyboxes, 226–227
- turning into ActionScript in Flash, 334
- using Flash Builder, 505–510

## **animation loops**

- building Wii navigation system, 413
- following game flow, 389

## **animation recorder, in PV3D, 512–515**

## **anti-aliasing, mipmapping reducing, 242**

## **anticipation, as animation principle, 28**

## **ArrayCollection**

- grabbing data from Twitter site, 563
- hooking up Flickr application, 566–567

## **arrays, in depth sorting, 20**

## **Artificial Intelligence (AI), adding to Pong game, 408**

## **AS3**

- creating XML in, 264–265
- depth sorting in, 20
- manipulating objects in Collada files, 188

## **AS3 Exporter, 202–204**

## **AS3Dmod, 360–362**

## **Ase, 179**

## **asymptotic zero. *See* vanishing point (asymptotic zero)**

## **asynchronous file operations, AIR, 307**

## **AsyncToken class, 282–283**

## **atoms, Molfile molecule viewer, 612–616**

## **attributes**

- particles in particle system, 227
- property access, 48
- XML, 264

## **augmented reality, 523–550**

- adding Jiglib physics to FLAR, 533–541
- adding PONG to FLAR, 542–548
- creating BasicView starter code, 527–531
- creating multiple markers, 541–542
- defined, 523
- generating own markers, 531–533
- making FLAR run faster, 549
- overview of, 523–526
- putting 3D model into FLAR, 549–550

## **Away3D**

- creating tube from Torus, 105–107
- naming Torus tube, 107
- porting in, 361–362
- porting torus from, 102–105

# **B**

## **backEnvMap, material for, 155–156**

## **background image**

- adding to 3D games for Wii, 368
- programming introState panel, 370–372
- programming winState panel, 372–373

## **background music, game flow, 387**

## **ball collision method, 546**

## **base state, 370**

## **BasicView**

- creating starter code for augmented reality, 527–531
- encapsulation and, 69
- example of, 69–70
- extending application using, 67
- JiglibFlash starter code using, 348–349
- undulating sphere using, 69–72

## **beginBitmapFill method, 134**

## **beginFill method, 117–118**

## **beginGradientFill method, 152–154, 595**

## **beginShaderFill method, 217**

## **Big and Small TV show, 460–461**

## **billboarding**

- adding physics to DisplayObject3D, 328
- Massive3D proximity based, 551
- overview of, 242–244
- saving CPU usage using, 460
- binary contrast, with Pixel Bender, 212**
- binary data, XML vs., 201**
- Bind to Data, news feed connection, 578**
- Bitmap property, MaterialObject3D class, 129**
- BitmapAssetMaterial, **136–141**
- BitmapColorMaterial, **141–142**
- BitmapData **class**
  - creating Google map wormhole, 607
  - creating video on primitive with, 143, 145
  - defined, 134
  - Flash Sandbox security issue, 255–259
  - flat shading with, 152–154
  - implementing light map in Flash CS4, 595
  - integrating particles and video, 240
  - saving bitmap data, 307–310
- BitmapFileMaterial **class**
  - bringing Pixel Bender into Flash, 216
  - building 3D Wall Carousel Course, 341–342
  - building nav system for 3D website, 474
  - overview of, 136–137
  - putting it all together, 137–141
- BitmapMaterial **class, 134–136**
- BitmapPixelMaterial **class, 216**
- bitmaps, 134–142**
  - adding skybox with motion, 225–226
  - BitmapAssetsMaterial, 136
  - BitmapFileMaterial. See BitmapFileMaterial class
  - BitmapMaterial, 134–136
  - BitmapViewportMaterial, 137–141
  - color and wireframe, 141–142
  - Pixel Bender instant, 207
  - putting it all together, 137–141
  - using, 134
- BitmapViewportMaterial, **137–141**
- black hat (deceptive) techniques, Web optimization, 483–484**
- Blender**
  - creating IPO curves in, 198
  - creating pyramid in CS4 using, 117–118
  - creating simple parser for, 202–203
  - as modeling application, 177
  - using drawTriangles to build orbiting ship, 599–606
  - using drawTriangles to build shaded TIE fighter, 594–599
- blimp (spherical) ellipsoids, 95–96**
- bloom brightness, Pixel Bender, 214**
- blow-up point (singularity), scaling equation, 7–8**
- bluetooth USB connector, Wii, 393**
- blur filter**
  - enhancing bump map, 163
  - Pixel Bender, 212
- bonds, Molfile molecule viewer, 615–617**
- Booleans**
  - changing motion in animation loops, 339
  - making Wii work with PV3D, 398
- bounce property, particle systems, 232–233**
- bounding boxes**
  - creating Pong game, 406
  - customizing, 375–376
  - limiting particle lifetime with, 233
- brightness, 157–160**
  - adding, 160–162
  - adding to color, Pixel Bender, 211
  - bloom, Pixel Bender, 214
  - point light, 147–148
- browser double-click problems, 337**
- buildPlane **method, cube prims, 85–87**
- bullets**
  - animating muzzle, 378–379
  - creating pool balls with imprint from, 380–381
  - creating tracer, 379
  - dropping imprint to optimize for Web, 391
- bump maps, 162–168**
  - creating, 162–163
  - overview of, 162
  - putting inside panorama, 163–168
  - using Fog and composite materials, 168–169
- Button\_Click **method, Flickr, 567**
- buttons**
  - 3D site layout in Photoshop, 500–501
  - navigation panel in Flash CS4, 494–495
  - navigation system for 3D website, 474–476
  - terrain editor, 296–297
  - twitter search application, 557, 563–565

## C

### camera

- adding physics to, 322–323
- adding video from Web cam to CS4 prim, 146–147
- creating CS4 super prim, 126

### **camera (continued)**

- creating perspectives, 323–324
- cruising stars using, 324–326
- defined, 40–41
- formulating `Papervision3D` class, 53
- using `BasicView`, 67–72
- view frustum, 38
- working with, 13–14

### **cancelState**

- adding in Flash Catalyst, 588
- defined, 370
- programming `winState` panel, 372–373

### **carousels**

- creating for rotation, 23–25
- using `ILayoutElement`, 619–620
- Wall Carousel Course. See Wall Carousel Course, 3D

### **cars**

- building interactive vehicles, 194–198
- building race track. See slot car racing game
- importing MD2 models, 181–182
- Wii virtual driving, 400–401

### **cartoon (cell) shading**

- overview of, 149
- using `CellMaterial`, 156–160

### **C/C++ libraries, Alchemy for, 251**

### **CDATA tags, XML, 264**

### **cell (cartoon) shading**

- overview of, 149
- using `CellMaterial`, 156–160

### **CellMaterial**

- creating shaders, 156–160
- updating with brightness parameter, 161

### **Center for Strategic and International Studies.**

- See **CSIS (Center for Strategic and International Studies) website**

### **changeList, instantiating RSO, 425–426**

### **chemical bonds, Molfile molecule viewer, 616–617**

### **Chinese Goodbye( ) method, GoodByeClass, 45–47**

### **Chinese Hello( ) method, HelloClass, 45**

### **class package, running PV3D in Flash as, 41–43**

### **class path, organizing classes in, 58**

### **classes**

- building own, 328–332
- building skybox, 373–375
- creating particles, 222
- customizing. See OOP (object oriented) class, building

- formulating. See PV3D (`Papervision3D`),
  - formulating class in
  - obtaining, 33–34

### **client side**

- adding database to Google maps, 280–284
- creating FMS application for, 419

### **clipping, 39**

### **clockTime( ) method, game scoreboard, 384**

### **clouds, 466–468**

### **CNN news reader, building, 577–583**

- adding style, 579–583
- binding data, 578
- configuring data, 578–579
- connecting service, 577–578
- overview of, 577

### **Cococo, 457–458**

### **code reuse, 98**

### **code view, programming panel containers, 370–373**

### **coding**

- animation, 9–12
- automatically in Flash Catalyst, 502
- Flash CS4 saving time, 610

### **COLLADA (COLLABorative Design Activity), 187.**

- See also **Collada parsers**

### **Collada, bringing 3D objects into PV3D with, 31–32**

### **Collada parsers**

- defined, 179
- difficulties importing files into PV3D, 202
- interactive objects and environments, 194–198
- introduction to, 187–188
- single and multiple objects with simple images, 188–191
- single object with multiple images, 191–194
- understanding, 204–206

### **collages, 465–466**

### **color**

- balance, in Pixel Bender, 211
- binary contrast in Pixel Bender, 212
- for low polygon models, 174
- optimizing for Web, 391
- for terrains, 289–293

### **Color material**

- adding to prim, 132–134
- `BitmapColorMaterial` vs., 141–142
- properties, 129

### **color property, ParticleMaterial, 222–223**

### **complement of image, 211**

### **composite materials, 168–169**

**concept mapping, recording 3D animation, 512–518****cones**

- creating hourglass from, 107–108
- creating pyramid from, 98
- generating, 93–95

**connections**

- CNN news feeds, 577–578
- to Flash Media Server, 419–420
- hooking up Flickr application in Flash Builder, 566–570
- tweeting application, using Flash Builder, 559–565

**container-sorting**

- creating carousel for navigation, 23–24
- overview of, 20–21

**controller, Wii, 393****coordinates**

- Flash 10 system of, 4–5
- Hello World using unprojected, 436

**Copy Motion, 29–31****copyPixels, 237****cosine, for flat shading, 153–154**

- `createBoundingBox` method, Jiglib, 374
- `createFloor( )` method, Jiglib Hello World, 351, 353–354

**createGun( ) method, 388****createHelloSphere( ) method, Jiglib Hello World, 351, 353****createmarkersXMLList, 267****createObjects function, 56****creationComplete( ) method, introState panel, 370****CRUD acronym, databases**

- creating custom client-side application, 280–284
- defined, 275
- saving data using SQLite, 312

**CS4. See Flash CS4****CSIS (Center for Strategic and International Studies) website, 462–483**

- building nav system, 474–476
- complete commented code, 476–483
- making clouds, 466–468
- making collage, 465–466
- making it 3D, 463–464
- making trees, 468–470
- overview of, 462–463
- placing panels with XML, 470–473
- using reflection, 464

**CSIS (Center for Strategic and International Studies) website, rebuilding in Flash CS4, 492–499**

- adding preloader, 499
- building navigation panel, 494–495
- comparing PV3D and Flash CS4, 493
- creating sliding display object, 495–498
- overview of, 492–493
- placing textarea on 3D panel, 493–494

**CSV (comma separated values), XML exporter, 203****cube prim**

- animating in CS4, 116–118
- creating custom puckered, 101–102
- creating custom Second Life Tree prim from, 108–110
- creating in CS4, 114–116
- rezzing to stage, 83–86
- transforming into pyramid, 97–98

**cubes**

- building interactive (face level), 332–333
- creating for Jiglib, 350

**curved plane prims**

- creating, 100–101
- creating puckered cube, 101–102
- creating video on Papervision3D prim, 144–145

**curved surfaces, for low polygon models, 174****custom classes**

- defined, 43
- extending with inheritance, 44–45
- placing custom methods underneath, 47–48
- writing, 43–44

**Custom Components, generating game states, 367****custom primitives. See primitives (prims), customizing****cutTriangleMesh method, slicing sphere, 89–91****cylinders**

- rezzing to stage, 91–93
- transforming into cones, 93–95

**D****DAE parser**

- defined, 179
- loading images onto object from, 199
- PV3D containing, 168
- steps for using, 199

**Daily Papervision, 462–463**

**dash, segmented line for track, 439–441**

**data provider, tweeting, 562–565**

**databases, 275–284**

installing test server, 276–277

integrating Flex with PHP. See PHP, automatically generating

map maker database, 277–278

saving data using SQLite, 312

understanding, 275

using Flex, 276

.db **extension, 311–312**

**Declaration of Independence port, 361–362**

**Delisle, Seb Lee, 19**

**deployment, AIR, 312**

**depth sorting (z-sorting), 20–21, 23–24**

**descendant operator (.), E4X, 265**

**development environment, FMS, 419**

**dictionaries, for particle trash, 241–242**

**digital certificates, AIR, 312**

**Disney, Walt, 28–29**

**display containers, 20–21, 23–24**

**DisplayObject3D class**

adding physics to, 326–328

building 3D Wall Carousel Course, 341–344

creating simple pendulum, 329–330

finding motion functions, 60

move commands, 197

overview of, 41

star cruising using cameras, 325–326

updating and adding brightness, 160–161

**document class, 57–58**

**documents, XML**

importing into Flex and Flash, 267–268

well-formed, 263–264

**DOM, storing recorded animations in, 515–516**

**donut radius, parametric equations, 16**

**dot syntax, E4X, 262–265**

**double pendulums, 330–331**

**double-sided plane prims, customizing, 110**

**double-sided property, MaterialObject3D**

**class, 129**

**dragging objects, 436–437**

**drawBitmap( ) method, pool balls with bullet**

**imprint, 380–381**

**drawing code, terrain editor in AIR, 294–295**

**drawParticle method, 223**

**drawRect method, creating carousel, 23**

**drawTriangles class**

building orbiting ship, 599–606

building shaded TIE fighter, 594–599

drawing in 3D using CS4, 114

rendering models in 3D, 594

taking to next level, 606

**drones, as military VR medium, 522–523**

**drop shadows, and Web optimization, 391**

## E

**E4X (ECMAScript)**

creating map makers, 283

incorporating XML into PV3D, 259

using, 264–265

**Eclipse, coding animation in, 9**

**ECMAScript. See E4X (ECMAScript)**

**edges, 35–36**

**ellipsoids, 95–96**

**Embed tag, 141, 435–436**

**emitter property, particle systems, 232**

**emotional (intuitive) programming, 246**

**emotional stimulation, 3D, 460**

**encapsulation**

overview of, 69–70

slicing image using, 238–239

**environment**

building in 3D, 460

converting site from 2D to 3D, 463–464

**Environment map (reflective) shading, 149, 155–156**

**EnvMapMaterial**

creating shaders, 155–156

PhongMaterial extending, 156

updating with brightness parameter, 161

**EnvMapShader, 162–163**

**evaluatePixel function, Pixel Bender,**

**209–210**

**EventDispatcher class, 3D games for Wii, 368**

**events**

overview of, 50

virtual mouse, 404

**exaggeration, as animation principle, 28**

**exclude faces parameter, cube prims, 86**

**exploding**

objects in Flash CS4, 233–237

in particle systems, 236

pool balls, 382–383

**exporting**

3D model from Photoshop, 511

Pixel Bender, 215

**exposure, adding in Pixel Bender, 211**  
**extra property, 3D site navigation, 476**  
**extrusions, creating with splines, 36**

## F

**face indices, 77, 79**

**face level mode, ISM, 332–337**

**face3d, sculpting prim, 312–315**

**faces**

- creating low polygon models, 174
- as element in object, 36

**factors of 2 principle, 242**

**far clip plane, view frustum, 38**

**File class, 307–310**

**file formats, importing into Photoshop, 511**

**FileReference class, 310–311**

**FileStream class, 307–310**

**FillAlpha property, Color material, 129**

**FillColor property, Color material, 129**

**filters**

- bringing Pixel Bender into Flash, 216
- optimizing for Web, 391
- Pixel Bender, 208–210

**final base code, formulating PV3D classes, 54–56**

**FindAll action token, 283**

**1st person perspective, camera, 323–324**

**FLARToolkit, 523–525**

**FLARToolkit, playing game in augmented reality**

- adding Jiglib physics to FLAR, 533–541
- adding PONG to FLAR, 542–548
- creating BasicView starter code, 527–531
- creating multiple markers, 541–542
- generating own markers, 531–533
- making FLAR run faster, 549
- putting 3D model into FLAR, 549–550

**Flash & Math, 591–593**

**Flash 10, 585–621**

- building 3D engine in 13 lines of code, 14
- optimizing PV3D engine for, 590–591
- overview of, 585
- rebuilding 3D pool game in Flash Builder, 586–591
- rotation in, 22–23
- saving to local PC using, 310–311
- using Flex or Flash components in, 345
- using ILayoutElement class, 620–621

**Flash 10, rendering 3D models, 591–620**

- creating 3D image ball, 591–593

- creating Google Map wormhole, 606–609

- creating Molfile molecule viewer, 610–617

- mapping single image onto octahedron, 617–620

- overview of, 591

- taking drawTriangles to next level, 606

- using drawTriangles to build orbiting ship, 599–606

- using drawTriangles to build shaded TIE fighter, 594–599

**Flash 3D**

- bringing Pixel Bender into, 215

- building 3D site, 460–461

- importing XML into, 266

- problems in PV3D, 344–345

- programming in Flex vs., 42–43

- sandbox security issue, 255–259

- using Alchemy, 251

- using Flex Embed tag in, 435–436

- using Google maps, 254

- validating XML structure before using in, 264

**Flash 3D, understanding**

- building Flash 10 engine in 13 lines with CS4, 14

- building Flash 9 engine. See Flash 9, building 3D engine in 18 lines of code

- coordinates in Flash 10, 4–5

- overview of, 3–4

- scaling, 28–32

- using rotation. See rotation, 3D

- using translation. See translation, 3D

**Flash 9, building 3D engine in 18 lines of code**

- adding camera, 13–14

- applying Thales Theorem, 5–6

- building Flash 10 engine in 13 lines with CS4, 14

- coding animation, 9–11

- deriving scaling equation, 6–8

- overview of, 5

- rendering to screen, 8–9

- running code, 11–12

- setting vanishing point (asymptotic zero), 12–13

**Flash Builder**

- adding game code, 589–590

- animating in 3D, 505–510

- building connectivity for tweeting, 559–565

- features, 503–504

- importing file from Flash Catalyst into, 559

- making 3D movies, 503–510

- new layout functionality of, 620–621

### **Flash Builder (*continued*)**

- rebuilding 3D pool game in, 586–591
- sending files from Flash Catalyst to, 503
- using Spark, 504–505

### **Flash Catalyst**

- adding 3D to Flickr menu, 566
- adding tweeting interactivity, 554–559
- creating code automatically, 501–502
- creating weather widget interface, 576–577
- hooking up components in Flickr application, 567–568
- importing file into Flash Builder, 503, 559
- laying it out in Photoshop, 500–501
- laying out game framework in, 586–587
- as next generation 3D software model, 585
- overview of, 500

### **Flash Communication Server, 417**

### **Flash CS4**

- building Flash 10 engine in 13 lines of code, 14
- building planetarium in, 273–275
- creating primitives in. See primitives (prims), creating in CS4
- creating stunning 3D websites, 463
- depth sorting in PV3D vs., 20–21
- early pioneer of, 120
- flat shading in, 594–597
- improving PV3D with, 72–73
- making 3D movies in, 492. See also CSIS (Center for Strategic and International Studies) website, rebuilding in Flash CS4
- making things explode in, 233–237
- native z component, 492
- putting Google map on, 259–262
- setting vanishing point, 14
- time and code savings of, 610
- video on prim in, 145–147

### **Flash Kit, 445**

### **Flash library, 113, 136**

### **Flash Media Server. See FMS (Flash Media Server)**

### **Flash XML graphics (FXG), 587**

### **flat shading**

- in Flash CS4, 594–597
- overview of, 149
- using `FlatShadeMaterial`, 152–154

### `FlatShadeMaterial`, **152–154, 161**

### **Flex**

- adding motion with, 60–65
- automatically generating PHP, 278–280
- building large-scale applications in, 345–346

- building PV3D Google maps in. See Google maps
- coding animation in, 9
- Embed tag, using in Flash, 435–436
- importing XML into, 266
- problems using in PV3D, 344–345
- programming in Flash vs., 42–43
- surfing to classes in, 246
- using with databases, 276
- validating XML structure before using in, 264

### **Flex 4 (Gumbo)**

- creating molecule viewer in, 610–617
- putting Google map on Web with, 259–262

### **Flex Builder, running PV3D in, 57–60**

### **Flex data wizard**

- automatically generating PHP with, 278–279
- creating custom client side application, 280–284
- PHP side, 280

### **Flickr menu, 565–570**

- adding 3D in Flash Catalyst, 566
- adding 3D in Photoshop, 565–566
- hooking it up in Flash Builder, 566–570
- overview of, 565

### **Flint particles**

- making it snow for Christmas, 245–246
- overview of, 249–251

### `floor` function, **Pixel Bender, 212**

### **FMS (Flash Media Server), 417–458**

- alternatives to, 456–458
- the big mistake, 421–422
- building race track, 437–456
- checking program performance, 422–423
- configuring development environment, 419
- connecting to, 419–420
- creating 3D multiplayer worlds using, 550
- dragging objects, 436–437
- free version of Red 5, 456–457
- getting started, 417–419
- overview of, 417
- running RSO Hello World, 423–436

### **FMS Administration Console, 422–423**

### **focal length**

- application of Thales Theorem, 5–6
- defined, 3
- deriving scaling equation, 7–8

### **focus, camera property, 53**

### **Fog Filter, 168–169**

### **folders, materials, 128**

### **follow-through, as animation principle, 28**

**Foucault pendulum, 331–332**  
**Fps, StatsView, 176**  
**Full Story link button, Flickr, 581**  
**functions, 43, 47–48**  
**FXG (Flash XML graphics), 587**  
**FXP file, 503**

## G

**game flow, 387–390**  
**game play, 389**  
**game score, 385**  
**game sight, 386**  
**game time, for game scoreboard, 383–384**  
**Geocoding, 253–262**  
     Flash Sandbox security issue, 255–259  
     overview of, 253–254  
     putting Google map on PV3D prim, 254–255  
     putting Google map on Web with Gumbo,  
         259–262  
**geodesic sphere prims, 99–101**  
**geometry.vertices property**  
     defined, 285  
     displacing vertices, 288–289  
     placing particles at vertices, 286  
     racking pool balls using, 381–382  
**getHeight method, displacing vertices,  
 288–289**  
**getPin( ) function, double pendulums,  
 330–331**  
**getPixel method, heightmaps, 288**  
**getRemote method, instantiating RSO, 424**  
**getTimer( ) method, StatsView, 176**  
**getTimer method, scoreboard, 383–384**  
**getXml( ) method, 266**  
**GoodByeClass, 45–46**  
**Google, Flash API. See Google maps**  
**Google 3D Warehouse, 511–512**  
**Google Earth, using KMZ, 180, 187**  
**Google Key, 256**  
**Google maps**  
     adding database to. See databases  
     creating wormhole in Flash 10, 606–609  
     Flash Sandbox security issue, 255–259  
     Geocoding and, 253–254  
     putting on Web with Flex 4 Gumbo, 259–262  
     putting PV3D prim on, 254–255  
**Google SketchUp. See SketchUp**  
**Gouraud shading, 149, 154–155**  
**GouraudMaterial, 154–155, 161**

**gradients**  
     adding color to terrain, 289–290  
     drawing code for terrain editor, 294–295  
     making trees for 3D site, 468–470  
**graphics, optimizing with mipmapping, 391**  
**gravity**  
     adding physics to DisplayObject3D,  
         326–328  
     particle systems, 232  
**grayscale**  
     adding key listeners to terrain editor,  
         299–306  
     converting image from color to, 211  
     heightmaps using, 287–288  
**Grden, John, 312–315**  
**Gumbo. See Flex 4 (Gumbo)**

## H

**hacking, in open source community, 102–103**  
**heightmap**  
     adding color to terrain, 289–293  
     overview of, 287–288  
     saving bitmap data, 307–310  
     turning Webcam into PV3D, 311–312  
     understanding, 287–288  
**Hello World**  
     Chinese Goodbye( ) method, 45–47  
     Chinese Hello( ) method, 45  
     extending class with inheritance, 44–45  
     JglibFlash, 351–356  
     programming in Flash vs. Flex, 42–43  
     running Papervision3D, 41–42  
     writing custom class, 43–44  
**Hello World, running RSO, 423–436**  
     instantiating RSO, 424–435  
     overview of, 423–424  
     unprojecting coordinates, 436  
     using Flex Embed tag in Flash, 435–436  
**HelloPlane. See BasicView**  
**hidden surface drawing, 37, 174**  
**highway, creating virtual, 441–442**  
**hitBall method, game play, 390**  
**hourglass prim, 107–108**  
**HTML, Web crawlers using, 484**  
**HTTP, running FMS, 417**  
**HTTP GET syntax, checking local weather, 571**  
**HTTPService**  
     building connectivity for tweeting in Flash  
         Builder, 562

### HTTPService (*continued*)

- creating connection for CNN news feed, 577–578
- creating custom client-side application, 280–284
- grabbing data from twitter site, 564
- hooking up Flickr application, 566

### human-readability, of Collada format, 187

## I

### if statement

- converting animation into ActionScript, 31
- creating cube prim, 85–87

### ILayoutElement class, 620–621

### Illustrator, producing FXG graphics in, 587

### image ball, 3D

- creating for rotation, 25–28
- creating in Flash 10, 591–593

### images

- adding to prim in CS4, 120–121
- building interactive cube (face level), 333–337
- creating heightmaps, 288
- creating low polygon models by optimizing, 174
- modifying source pixels with point processes, 210–215
- processing in Pixel Bender, 210–215
- writing Pixel Bender filter, 209–210

### import statements

- building wire frame sphere, 51
- obtaining, 50
- for properties, 50
- putting materials together in application, 137

### importers, PV3D. See parsers

### importing files

- from Flash Catalyst into Flash Builder, 559
- XML into Flex and Flash, 266

### indices, creating cube based on, 114–116

### Influxis

- getting Massive3D using, 550–551
- getting started with Flash Media Server, 417–418
- preventing insecurity of swf files, 421–422

### inheritance, 44–47

### init( ) method, 387

### init2D( ) method, 388

### init3D( ) method, 388

### initialization, coding animation using, 10

### initP1 panel, introState panel, 370–371

### initP2 panel, winState panel, 372–373

### initPV3D function

- final base code, 56
- following game flow, 387–388
- formulating PV3D class, 51

### input command, Pixel Bender, 209

### inside faces parameter, cube prim, 86

### installer files, AIR, 312

### instantiating RSO, FMS Hello World, 430–435

- assigning Sync event, 425
- creating application, 426–435
- using changeList, 425–426
- using getRemote, 424
- using setProperty, 424–425

### integrating environments

- Flash Media Server. See FMS (Flash Media Server)

- Wii-PV3D game design, 404

### interactive objects, Collada, 194–198

### Interactive property, MaterialObject3D class, 129

### Interactive Scene Manager. See ISM (Interactive Scene Manager)

### InteractiveScene3D object, 332

### InteractiveScene3DEvent listener, 312–315

### interactivity

- building nav system for 3D site, 474–476
- with objects in PV3D. See ISM (Interactive Scene Manager)
- slicing image using, 239–240

### interface, weather widget, 576–577

### internal access modifier, 48

### InterPOlation (IPO) curves, Blender animation, 198–199

### introState

- adding in Flash Catalyst, 588
- defined, 370
- programming panel containers, 370–372

### intuitive (emotional) programming, 246

### IPO (InterPOlation) curves, Blender animation, 198–199

### IR data, paddle motion, 414–415

### ISM (Interactive Scene Manager), 332–341

- browser double-click problems, 337
- building 3D Wall Carousel Course, 341–344
- building interactive cube (face level), 332–337
- building interactive particles (object level), 338
- clicking on particles, 339–340
- overview of, 332
- rolling over x, y, z gizmo axes, 340–341
- rotating small sphere, 338–339

**J****JiglibFlash**

- adding objects, 349–351
- adding physics to FLAR, 533–541
- adding simple preloader, 360
- BasicView starter code, 348–349
- building example viewer, 356–359
- creating Pong. See Pong game, with Jiglib
- Hello World, 351–356
- overview of, 348
- pool shooting game, 366–367
- WOW vs., 346

JiglibLoaderApp **modules folder, 356–359**

**Jing, using, 565**

**JPEG encoder, saving bitmap data, 307–309**

**K****key listeners**

- building Wii navigation system, 411–413
- creating terrain editor, 299–306
- sculpting prim using face3d, 312–315

**keyboard listener, setting up for game, 389**

**KMZ parser**

- defined, 180
- getting Collada files out of, 201
- overview of, 187

**L**

**LAMP test server, installing, 276–277**

**latitude, Google map wormhole, 606–609**

**layout**

- adding 3D to Flickr in Photoshop, 565–566
- adding tweeting interactivity in Flash Catalyst, 555–556
- Flash Builder's new functionality, 620–621
- game framework in Flash Catalyst, 586–587
- for twitter search application in Photoshop, 554

**LEDJAM website, 463–464**

**Lee-Delisle, Seb, 244–246, 405, 461**

**Leskinen, Petri, 120**

**Letter3D material, 246–247**

**libraries, Collada document, 195, 204–206**

**lifetime, limiting particle, 233**

**light maps**

- CellMaterial, 156–160
- creating, 152–153

- FlatShadeMaterial, 153–154
- implementing flat shading in Flash CS4, 595–596

PhongMaterial, 156

**light source, in Pixel Bender, 217–218**

**light vectors, 148**

**lightColor variable, flat shading, 595**

**lighting**

- adding brightness, 160–162
- creating multiple light source with Pixel Bender, 217
- creating shade materials. See shaders
- understanding, 147–148

**LineAlpha property, Wireframe material, 129**

**LineColor property, Wireframe material, 129**

**lines**

- avoiding for low polygon models, 174
- creating racing track, 439
- creating simple pendulums, 329–330
- making bullet tracer, 379
- removing blur for Web optimization, 391

**lineStyle method, animating cube in CS4, 117–118**

**LineThickness property, Wireframe material, 129**

**linking, creating steering mechanism, 195–196**

**List box, twitter search application, 556–559**

**listeners**

- coding animation, 10–11
- creating terrain editor with key, 299–306
- sculpting prim using face3d key, 312–315
- slicing images, 239–240

**LiveSupport, 418**

**local file system, AIR using, 307**

**longitude, Google map wormhole, 606–609**

**looping**

- coding animation, 11
- formulating PV3D class, 51
- rendering PV3D class, 54

**loseState**

- adding in Flash Catalyst, 588
- defined, 370
- programming winState panel, 372–373

**Lynda.com tutorial video service, for Photoshop, 461**

**M**

**MAMP test server, installing, 276–277**  
**map, creating site, 461–462**

## **map maker database, 277–278, 283**

### **mapping**

- single image onto octahedron, 617–620
- using `drawTriangles` to build orbiting ship, 599–606

### **markers**

- bringing media into, 267–268
- feeding XML data into popup, 267
- generating multi-FLAR, 541–542
- generating own, 531–533

## **mass concept, physics, 321–322, 326–328**

## **Massive3D, 418, 550–551**

## **material editor, 178, 179**

### **MaterialObject3D class, 129**

### **materials, 127–169**

- adding brightness, 160–162
- adding skybox with motion, 225–226
- adding to prim, 130–134
- applying textures, 127–128
- bitmaps. See bitmaps
- bump maps, 162–168
- creating custom Second Life Tree prim, 109–110
- for cube vs. other prims, 83
- defined, 40–41
- Fog and composite, 168–169
- movie clips, 142–147
- overview of, 127
- types of, 130
- understanding light, 147–148
- using shaders. See shader materials
- using simple, 128

### **MaterialsList class**

- adding multiple materials with, 191
- adding skybox with motion, 225–226
- building interactive cube (face level), 333–337
- creating cube prim, 85–87
- loading images onto object from, 199–200

## **Math.random( ) \* 0xfffff method, 17**

### **Matrix3D object**

- animating cube in CS4, 116–117
- applying transformations all at once, 15, 125–126
- container-sorting, 21
- creating CS4 super prim, 120–125
- defined, 4
- drawing using CS4, 114

## **Max3DS parser, 180**

## **Maya application, 178**

## **MD2 parser, 180–187**

- animation in, 186–187
- building ray gun, 377–378
- creating files with MilkShape, 185–186
- defined, 180
- handling multiple models, 184–185
- model pack, 181–184
- putting into PV3D, 180–181

## **Media Interactive Server, 417**

## **Media Streaming Server, 417**

### **memory usage**

- gauging with StatsView, 176
- handling particle trash, 240–242

### **methods**

- creating, 47–48
- defining scope for, 48–49
- drawing in 3D using CS4, 114

## **military virtual reality, 521–523**

## **MilkShape, 185–186, 377**

## **mind-mapping, 512**

### **mipmapping**

- billboarding surfaces to look 3D, 242
- creating low polygon models, 174
- exporting Photoshop collage for 3D website using, 465
- optimizing for Web with, 391
- overview of, 244

## **modal control, 3D games for Wii, 367–368**

## **model packs, MD2, 181–184**

## **modeling program, PV3D, 285–293**

- modeling terrains, 287–293
- overview of, 285
- placing particles at vertices, 286–287
- sculpting prim using face3d, 312–315

## **models, 3D, 173–220**

- bringing down count, 174–175
- bringing into FLAR, 549–550
- choosing application, 176–179
- gauging performance in PV3D, 175–176
- importers for PV3D. See parsers
- overview of, 173–174
- processing images. See Pixel Bender
- writing own parsers, 201–206

## **modolo function, 439–442**

### **modules**

- adding simple preloader, 360
- building viewer in Jiglib, 356–359

## **molecule viewer. See Molfile molecule viewer**

## **Molfile molecule viewer, 610–617**

- creating `Atom` class, 612–615
- creating chemical bonds, 616–617

- creating molecules and bonds, 615
  - Molfile parser, 610
  - overview of, 610
  - placing atoms, 615–616
  - motion, adding, 60–65**
  - motion editor**
    - converting animation into ActionScript, 29–31
    - defined, 29
    - video tutorial for, 29
  - motion paths, Spark, 516–518**
  - mouse**
    - adding paddle motion, 414–415
    - adding virtual, 402–403
    - in screen coordinates, 436
  - mouse listener, games, 389**
  - move commands, DisplayObject3D, 197**
  - `moveForward` **command, skybox motion, 227**
  - movie clips, 142–147**
    - building interactive cube (face level), 333
    - connecting to `Particle3D` class, 229–230
    - exploding particle system using, 234–235
    - Google map wormhole using, 607–608
    - importing into PV3D, 142
    - optimizing for Web, 391
    - using `MovieAssetsMaterial`, 142
    - using `MovieMaterial` class, 142
    - using `VideoStreamMaterial`, 143
    - video on CS4 prim, 145–147
    - video on PV3D prim, 144–145
  - `MovieAssetMaterial`, **142, 333**
  - `MovieMaterial` **class**
    - building interactive cube (face level), 333
    - creating pool balls with bullet imprint, 380–381
    - Flash Sandbox security and, 255–259
    - putting Google map on PV3D prim, 254–255
    - using, 142
  - movies, making 3D, 491–519**
    - building PV3D studio, 518–519
    - recording 3D animations, 512–518
    - technological changes and, 491–492
    - using Flash Builder, 503–510
    - using Flash Catalyst, 500–503
    - using Flash CS4, 492. *See also* CSIS (Center for Strategic and International Studies) website, rebuilding in Flash CS4
    - using Photoshop, 511–512
  - muzzle, animating, 378–379**
  - MXML application**
    - FXG as subset of, 587
    - programming in Flash vs. Flex, 43
    - running PV3D in Flex as, 41–43
  - myAngle+i/20, 17–18**
  - `myDisplay` object **name, carousels, 23**
  - `mySlideLayer` **movie clip, 495–498**
  - MySQL database**
    - creating map maker, 277–278
    - defined. *See also* databases
    - generating PHP for. *See* PHP automatically generating
    - installing test server, 276–277
  - `myTerrain` **vertices, 288–289**
- ## N
- Name property, MaterialObject3D class, 129**
  - naming conventions**
    - custom classes, 43–44
    - model in 3DSMax, 195
    - new prims, 96
    - Pinched Sphere, 97
  - `NaN` **data, Wiimote, 414–415**
  - National Treasure movie, 361–362**
  - navigation**
    - 3D site, 474–476
    - converting site from 2D to 3D, 474–476
    - in Flash CS4 for 3D movies, 494–495
    - laying out 3D site in Photoshop, 500–501
    - rotating small sphere with ISM, 338–339
    - star cruising using cameras, 324–326
    - using view states, 368–369
    - Wii, 411–413
  - near clip plane, view frustum, 38**
  - net connection**
    - connecting to FMS, 419–420
    - video on CS4 prim, 145
  - `NetStream` **class, 144–146**
  - news feed. *See* CNN news reader, building**
  - Newtonian Mechanics, 322**
  - nochump folder, 34**
  - nodes, XML, 264**
  - NOMAD, 268**
  - nonconvex rigid body physics, 348. *See also* JiglibFlash**
  - normal vector, 83**
  - `null` **command, particle trash, 241**
- ## O
- object level mode, ISM, 332, 338–341**
  - objects**

### **objects (continued)**

- adding in JiglibFlash, 349–351
- defined, 35, 40
- dragging, 436–437
- interactivity of. See ISM (Interactive Scene Manager)
- optimizing for Web, 392
- overview of, 36
- in world coordinates, 436

### **octahedron, mapping single image onto, 617–620**

- `OnFrame( )` method, **StatsView**, 176
- `onRenderTick` method, **70–72, 547–548**
- OOP (object oriented) class, building, 43–50**
  - adding custom properties, 49–50
  - creating methods, 47–48
  - defining scope, 48–49
  - demonstrating inheritance, 45–47
  - extending with inheritance, 44–45
  - overview of, 43–44

### **open source community, benefits of hacking in, 102–103**

### **Open Source Flash, 456–457**

### **optimizing for Web**

- avoiding black hat techniques, 483–484
- building 3D games, 390–392

### **orbiting ship, building, 599–606**

### **org folder, 34**

### **oscbright, 161–162**

### **oscillators, and Foucault pendulum, 331–332**

### **outCoord property, Pixel Bender, 209**

### **output command, Pixel Bender, 209**

### **overlays**

- creating pool balls with bullet imprint, 380–381
- or Flex or Flash components, 345

### **override statement**

- `ChineseGoodbye( )` method, 46–47
- using `BasicView`, 67–69

## **P**

### **Package Explorer panel, Flash Builder, 503**

### **paddle methods, 546**

### **Painter's Algorithm, 37**

### **paletteMap**

- adding color to terrain, 289–293
- and `CellMaterial`, 156–160

### **panels, 3D site**

- building nav system, 474–476
- collapsing in Flash Catalyst, 557–559

- creating sliding movie clip, 495–498
- creating weather widget interface, 576–577
- laying out in Photoshop, 500–501
- placing textarea on 3D panel in Flash CS4, 493–494
- positioning on stage with XML, 470–472
- toggling, 472–473

### **pano objects**

- adding skybox with motion, 225–226
- creating, 192–193

### **panoramas**

- bringing into PV3D, 193–194
- creating, 163
- creating cubic, 191–192
- creating pano object, 192–193
- putting bump map inside, 163–168

### **Paperplane primitive, 76**

### **paraboloid, puckered cubes, 101–102**

### **parameter command, Pixel Bender filter, 210**

### **parameterized splines, racing tracks, 438–441**

### **parameters**

- adding to Pixel Bender, 210
- Perlin noise, 298

### **parametric equations**

- creating CS4 super prim, 120–125
- image ball generated from, 26–28
- mapping single image onto octahedron, 619–620
- overview of, 15–16
- for torus, 17–19, 103–105
- for tubes, 105–107

### **parametric particle system (Torus worm), 17–20**

### **parsers, 179–201**

- 3DSMax, 204
- Blender, 202–203
- Collada. See Collada parsers
- KMZ, 187
- MD2. See MD2 parser
- Molfile, 610–612
- overview of, 201–202
- SketchUp, 200–201
- types of, 179–180
- using binary data vs. XML, 201

### **particle systems, 221–252**

- building interactive particles (object level), 338–341
- building modeling program, 286–287
- defined, 221
- Flint particles, 249–251
- key concepts, 232–233

- Plug-in media, 244–248
- properties of, 221–222
- in PV3D, 222–227
- treating multiple models like, 184–185
- using Alchemy for more bang, 251
- particle systems, creating generic CS4 particle class, 227–244**
  - ActionScript main code for, 230–231
  - building `Particle3D` class, 228–229
  - connecting movie clip to, 229–230
  - exploding objects, 233–237
  - handling trash, 240–242
  - integrating video, 240
  - key concepts, 232–233
  - lifetime or bounding box, 233
  - making 2D surfaces look 3D (billboarding), 242–244
  - mipmapping, 244
  - overview of, 227
  - slicing image, 237–240
- `Particle3D` **class**
  - billboarding surfaces to look 3D, 242–244
  - building, 228–229
  - connecting movie clip to, 229–230
  - creating, 228–229
  - handling particle trash, 240–242
  - key particle concepts, 232–233
  - making things explode (Flash CS4), 233–237
- `ParticleBlowUp` **class, 233–237**
- `particleClick` **snippet, 340**
- `ParticleField` **class, 222, 269–273**
- `particleLife` **variable, 233**
- `ParticleMaterial` **class**
  - building planetarium in PV3D, 269–273
  - creating particles in PV3D, 222
  - putting stars in skybox, 223–227
- `Particles` **class, 222–223**
- paths, creating with splines, 36**
- pendulums, building physics classes, 328–332**
- performance**
  - Flash Media Server, 422–423
  - gauging PV3D, 175–176
  - making FLAR run faster, 549
  - optimizing PV3D for Flash 10, 590–591
- Perlin noise**
  - creating terrain editor, 298–299
  - defined, 298
  - using `drawTriangles` to build shaded TIE fighter, 594
- perlin plane, 67–68**
- perspective concept, physics, 321–324**
- perspective scaling**
  - building 3D engine. See Flash 9, building 3D engine in 18 lines of code
  - creating illusion of depth, 6
  - derived from Thales Theorem, 3
  - rendering to screen, 8–9
- Phong shading**
  - creating bump map, 162–163
  - defined, 149
  - overview of, 156–160
- `PhongMaterial`, **156, 161**
- Photoshop**
  - adding 3D to Flickr menu, 565–566
  - building Jiglib bounding box, 374
  - creating layout for tweeting, 554
  - CS4 engine, 511–512
  - laying out 3D site, 500–501
  - learning for 3Dsites, 461
  - making collage in, 465–466
  - producing FXG graphics, 587
- PHP**
  - creating map maker database, 277–278
  - defined, 275
  - installing test server, 276–277
- PHP, automatically generating, 278–284**
  - creating custom client side application, 280–284
  - defined, 276
  - overview of, 278
  - PHP side, 280
  - using Flex data wizard, 278–280
- `phpMyAdmin`, **map maker database, 277–278**
- physics, 3D, 321–364**
  - 3rd party engines, 346–348. See also JiglibFlash
  - adding Jiglib to FLAR, 533–541
  - adding to camera, 322–323
  - adding to `DisplayObject3D` class, 326–328
  - AS3Dmod, 360–362
  - building own classes, 328–329
  - creating 3D wall course, 341–346
  - creating camera perspectives, 323–324
  - cruising stars using cameras, 324–326
  - double pendulum, 330–331
  - Foucault pendulum, 331–332
  - interacting with objects in PV3D. See ISM (Interactive Scene Manager)
  - overview of, 321
  - simple pendulum, 329–330

**physics, 3D (continued)**

- torsion pendulum, 331
- understanding guts of, 321–322

**Pinched Sphere, 96–97**

**pinning, creating double pendulums, 330–331**

**pipe command, cubes, 86–87**

**Pixel Bender, 206–218**

- adding parameters, 210
- adding to PV3D, 215–217
- animating light source, 218
- bringing into Flash, 215
- creating multiple light source, 217–218
- getting started, 207–208
- Pixelero blog resource for, 606
- problem optimizing PV3D for Flash 10, 591
- rendering large particle sets, 251
- typical point processes, 210–215
- understanding, 206–207
- writing filter, 208–210

**Pixelero blog, 606**

**pixels, modifying source, 210–215**

**plane prim**

- constructing, 78–79
- CS4 native, 114
- curved, 100–101
- custom double-sided, 110
- rezzing to stage, 81–83

**Plane3D method, slicing sphere, 89–91**

**planes**

- 3D Wall Carousel Course, 341–344
- Jiglib Hello World, 351–356
- JiglibFlash, 350
- making clouds, 467–468
- making trees, 468–470

**planetarium**

- building in Flash CS4, 273–275
- building in PV3D, 268–273

**playState, Flash Catalyst, 588**

**playState panel, 370–371**

**Plug-in media**

- Big and Small* TV show, 460–461
- making snow for Christmas, 245–247
- overview of, 244–248
- scanning, 247–248

**PNG encoder, 307–309**

**png files, Second Life tree prim, 109–110, 111–112**

**point light**

- constructing, 147–148
- creating shaders. See shaders

Pixel Bender simple, 213

purpose of, 151

**point processes, Pixel Bender, 210–215**

**point sprites, saving CPU usage, 460**

**polar octahedron derivation, 617–618**

**polygons**

- bringing down count when modeling, 173–174
- defined, 35
- making 2D surfaces look 3D, 242–244
- modeling for PV3D, 173
- reducing by removing hidden surfaces, 37
- rendering using Painter's Algorithm, 37–38

**polymorphism**

- creating particles in PV3D, 222
- slicing image using, 238–239

**PONG, adding to FLAR, 542–548**

- adding ball collision method, 546
- adding paddle methods, 546
- adding Pong properties, 542–543
- adding PONG to FLAR, 543–545
- adding textboxes to keep score, 545–546
- inserting code into `onRenderTick`, 547–548
- renaming file, 542
- transferring Embed assets, 542
- transferring Lively folder, 542

**Pong game, with Jiglib, 404–425**

- adding AI, 408
- adding Jiglib, 406–407
- adding mouse-controlled paddle motion, 414–415
- adding reflectivity and skybox, 411
- building bounding box, 406
- building navigation system, 411–413
- creating textboxes, 410–411
- making game levels and scoring system, 408–410
- moving at constant velocity, 408
- overview of, 404–405

**pool shooting game**

- adding background image, 368
- adding Wii to, 402–404
- building custom skybox, 373–375
- building Jiglib bounding box, 375–376
- creating animated muzzle, 378–379
- creating pool balls with bullet imprint, 380–381
- creating ray gun, 377–378
- creating tracer, 379
- exploding pool balls, 382–383
- following game flow, 387–390
- game sight, 386

- game states, 367–370
- optimizing for web, 390–392
- overview, 366–367
- programming panel containers, 370–373
- racking pool balls, 381–382
- scoring balls, 383–386
- pool shooting game, rebuilding in Flash Builder, 586–591**
  - adding code, 589–590
  - adding game states in Flash Catalyst, 588–589
  - laying out framework, 586–587
  - overview of, 586
- Popup Manager, generating game states, 367**
- popups, feeding XML data into, 267–268**
- portability, Second Life Tree prim in CS4, 112**
- porting, 102**
- position concept, physics, 321–322**
- preloaders**
  - converting site from 2D to 3D, 476
  - creating 3D movies in Flash CS4, 499
  - implementing `BitmapAssetsMaterial` in, 136–137
  - simple `JiglibFlash`, 360
- Presenter Chat, 418**
- primitives (prims)**
  - adding materials to, 130–134
  - defined, 56
  - exploring guts of, 76–79
  - putting Google map on PV3D, 254–255
  - sculpting using `face3d`, 312–315
  - understanding, 75–76
  - using `Matrix3D`, 125–126
  - using `VideoStreamMaterial` to play video on, 143
  - video on CS4, 145–147
  - video on Papervision3D, 144–145
- primitives (prims), creating in CS4, 110–125**
  - animate cube, 116–118
  - create cube, 116
  - cube, 114–116
  - overview of, 110–111
  - plane, 114
  - pyramid, 118–119
  - Second Life Tree, 111–114
  - super prim, 120–125
- primitives (prims), customizing, 96–110**
  - curved plane, 100–101
  - double-sided plane, 110
  - geodesic, 99–100
  - hourglass, 107–108
  - overview of, 96
  - Pinched Sphere, 96–97
  - puckered cube, 101–102
  - pyramid, 97–99
  - Second Life Tree, 108–110
  - torus, 102–105
  - tube, 105–107
- primitives (prims), rezzing to stage, 79–96**
  - cone, 93–95
  - cube, 83–87
  - cylinder, 91–93
  - making ellipsoids, 95–96
  - overview of, 79
  - plane, 81–83
  - prim template code ( wireframe material), 80–81
  - slicing sphere, 89–91
  - sphere, 87–89
  - wireframe sphere. See spheres, wireframe
- primitives folder, 76**
- private access modifier, 48–49**
- processing images. See Pixel Bender**
- projection plane (viewport), Thales Theorem, 5–6**
- `projectVectors` method, drawing in 3D using CS4, 114
- properties**
  - adding PONG to FLAR, 542–543
  - creating `Particle3D` class, 228
  - custom, 49–50
  - defined, 43
  - defining scope, 48–49
  - particle, 227
  - particle systems, 232–233
  - putting stars in skybox, 223–227
  - simple materials, 129
  - Wireframe material, 129
- protected access modifier, 48**
- prototypes, military virtual reality, 522**
- proximity based billboarding, Massive3D, 551**
- PSD files, 554–559**
- public access modifier, 48–49**
- PV3D (Papervision3D)**
  - adding Pixel Bender to, 215–217
  - building modeling program, 285–293
  - building planetarium, 268–273
  - building studio, 518–519
  - creating Second Life tree, 112
  - CS4 vs., 493
  - CS4 vs. version 2 of, 259
  - defined, 3

### **PV3D (Papervision3D) (continued)**

- depth sorting, 20–21
- gauging performance, 175–176
- optimizing for Flash 10, 590–591
- particle systems, 222–227
- using Collada Files, 31–32
- using Flint, 249–251

### **PV3D (Papervision3D), formulating class in, 50–57**

- building custom class. See OOP (object oriented) class, building
- camera, 53
- creating pyramid prim, 98
- final base code, 54–57
- overview of, 50–52
- renderer, 54
- scene, 53–54
- viewport, 52–53

### **PV3D (Papervision3D), getting started, 33–73**

- adding motion, 60–66
- application elements, 39–41
- building OOP custom class, 43–50
- building Papervision3D class, 50–57
- hidden surface drawing, 37
- history of, 33
- incorporating CS4, 72–73
- obtaining, 33–34
- Painter's Algorithm, 37–38
- running applications, 57–60
- running four different ways (Hello World), 41–43
- shapes vs. objects, 35–36
- simulating 3D objects in ActionScript, 34–35
- using `BasicView` (Hello Plane), 67–72
- view frustum (culling and clipping), 38–39

### **PV3D plug-in, Jiglibflash, 348–349, 373–375**

### **pyramids, creating, 97–98, 118–119**

### **Python, 202–203**

## **Q**

### **quadrant render engine, 378**

### **Quake II engine. See MD2 parser**

### **quality, stage, 66**

### **Quantum Mechanics, 322**

## **R**

### **race track. See slot car racing game**

### **racking pool balls, 381–382**

### **radio buttons, creating, 295**

### **rawChildren tag, using MXML with, 43**

### **ray gun, modeling, 377–378**

### **ray tracing, 151**

### **realism, with physics. See physics, 3D**

### **recording 3D animations, 512–518**

- animating Spark, 516–518
- creating animation recorder in PV3D, 512–515
- overview of, 512
- storing results using DOM, 515–516

### **Red 5, 456–457**

### **reflection**

- 3D site, 464
- collage for 3D site, 465
- Pong game, 411
- sliding movie clip, 496–498

### **ReflectionView, 464**

### **regenerator property, particle systems, 232**

### **Relativistic Mechanics (special relativity), 322**

### **Remote Shared Objects (RSO), 423**

### **removeChild method**

- handling particle trash, 241
- removing models from stage, 184

### **renderers**

- defined, 40–41
- formulating PV3D class, 54
- using `BasicView`, 67–72
- using Flint in PV3D, 249–251

### **rendering. See also Flash 10, rendering 3D models**

- adding motion, 61–62
- defined, 36
- hidden surface drawing for, 37
- overview of, 7–8
- using Painter's Algorithm, 37

### **RenderStatistics class, StatsView, 176**

### **restitution, Pong game, 406–407**

### **resultHandler method, 281–284**

### **rezzing prims. See primitives (prims), rezzing to stage**

### **rooms, building 3D, 460**

### **Rotate3D Spark effect, Flash Builder, 506–510**

### **rotation, 3D, 21–28**

- adding motion using, 60–62
- adding to skybox, 226–227
- building interactive particles (object level), 338–339
- creating carousel, 23–25
- creating image ball, 25–28
- creating sliding movie clip, 496–498
- Flash 10, 22–23
- with Flash Builder, 506–510

Flash CS4 tool, 492  
 Matrix3D class and, 4, 125–126  
 overview of, 21  
 planes, 81–83  
 spheres, 87–88  
 rotationX **command**, 22–23  
 rotationY **command**, 22–23  
 rotationZ **command**, 22–23  
**round tripping, Flash Catalyst/Flash Builder**, 503  
**RSO (Remote Shared Objects)**, 423  
**RSS Feed, checking local weather**, 571  
**RTMP**, 417, 419, 421–422  
**Rubik's cube**, 285–286  
**Runge-Kitta**, 331  
**running code**, 11–13

## S

**Sandbox, and security**, 255–259

**saving projects**

in 3DSMax, 178  
 in Blender, 177  
 before upgrading, 34

**scale modes, stage**, 66

**scaling**, 28–32

adding motion with, 60, 63–65  
 defined, 28  
 Matrix3D class handling, 4, 125–126  
 mipmapping reducing, 242  
 with Thales Theorem. See Thales Theorem  
 turning timeline animation into ActionScript,  
 29–31  
 using Collada files, 31–32  
 Walt Disney's Rules of animation, 28–29

**scanning**, 247–248

**scenes**

defined, 39–41  
 formulating PV3D class, 53–54  
 optimizing programmatically, 176  
 putting materials together in, 138–139  
 using `BasicView`, 67–72

**scope (access modifiers)**, 48–49

**scoring systems**

adding PONG to FLAR, 545  
 game scoreboard, 383–386  
 Pong game, 408–410

**screen (x, y) coordinates**, 436–437

**Script Tag application, and MXML**, 43

**scroll bar, twitter search application**, 555–556

**scroll policies, Web optimization**, 391, 462

**search**

adding to twitter, 557  
 grabbing data from twitter site, 563–565

**search engine result pages (SERPs)**, 483

**Second Life tree primitives**

creating, 75–76  
 creating in CS4, 111–114  
 customizing, 108–110

**2nd person camera perspective**, 323–324

**security, and Flash sandbox**, 255–259

**segments, spline**, 35–36

**self-signed certificates, deploying AIR**, 312

**sensor bar, adding Wii to game**, 401–402

**SEO (Search Engine Optimization)**

ever-changing art of, 488  
 getting pages higher, 483  
 overview of, 484  
 using HTML, 484  
 using `SWFObject`, 485–488

**SERPs (search engine result pages)**, 483

**server side, FMS development**, 419

**services, adding Web**, 553–583

adding 3D to Flickr menu, 565–570  
 CNN news feed. See CNN news reader, building  
 overview of, 553  
 tweeting. See twitter search application  
 Yahoo. See Yahoo Web services, mining

`setPixel` **method, terrain editor**, 293–295

`setProperty`, **instantiating RSO**, 424–425

`setSpringCamera( )` **method, Jiglib Hello  
 World**, 351, 353

`setStage` **method**, 70–72

`setUpCamera( )` **method, game flow**, 388–389

`setUpListeners( )` **method, game flow**, 389

**Seven Revolutions project**

building. See CSIS (Center for Strategic and  
 International Studies) website  
 overview of, 462–463  
 rebuilding in CS4. See CSIS (Center for Strategic  
 and International Studies) website, rebuilding  
 in Flash CS4

**shader materials**, 148–160

`CellMaterial`, 156–160  
 code template, 150–151  
 coding, 149  
`EnvMapMaterial`, 155–156  
`FlatShadeMaterial`, 152–154  
`GouraudMaterial`, 154–155  
 overview of, 148–149  
`PhongMaterial`, 156

## shaders

- bump map, 162–163
  - low polygon model, 174
  - Pixel Bender, 208–210
  - as processor intensive, 168–169
  - updating, 161–162
  - using `drawTriangles` to build shaded TIE fighter, 594–599
- shapes, 35**
- sharpen filter, Pixel Bender, 213**
- shift command, particle trash, 241**
- short cuts, Blender, 177**
- Show Gun checkbox, scoring balls, 384–385**
- showTime method, games, 384, 389**
- sight, game, 386**
- silly string program, Wii, 397–399**
- Simple Cube class, FLARToolKit, 524**
- simple materials (or just materials), 129**
- simple pendulums, 329–330**
- Simple Pixelate, Pixel Bender, 212**
- simple point light, Pixel Bender, 213**
- simulators, military VR, 522**
- singularity (blow-up point), scaling equation, 7–8**
- site maps, 3D sites, 461–462**
- SketchUp**
- Google 3D Warehouse and, 511–512
  - as modeling application, 178–179
  - overview of, 200–201
  - parser, 180
- SketchUp Pro, 201**
- SketchUpCollada class, 180**
- skyboxes**
- adding reflectivity to, 411
  - building custom, 373–375
  - dropping for Web optimization, 391
  - putting stars in, 223–227
- slicing**
- images into particles, 237–240
  - spheres, 89–91
- slide layer, clouds for 3D site, 468**
- sliders**
- adding to terrain editor, 296
  - creating torsion pendulums with, 331
- sliding display object, 3D movies, 495–498**
- slot car racing game, 437–456**
- adding FMS to, 444–456
  - constructing virtual highway, 441–442
  - creating track, 438–441
  - having wreck, 442–444
  - overview of, 437–438

- turning code into game, 456
- Smooth property, MaterialObject3D class, 129**
- snow, particle effect, 247–248**
- snow systems, 244–246**
- sorting, depth, 20–21**
- Sothink SWF Decompiler, 421–422**
- space bar, Blender, 177**
- Spark**
- animating, 516–518
  - overview of, 504–505
  - using FXG graphics in, 587
- special relativity (Relativistic Mechanics), 322**
- specularLevel variable, flat shading, 595**
- speed parameter, Wii virtual driving, 401**
- sphereMorphOver method, sculpting prim, 312–315**
- spheres**
- building interactive particles (object level), 338–339
  - instantiating for Jiglib Hello World, 351–356
  - for JiglibFlash, 351
- spheres, wireframe**
- adding motion, 60–66
  - adding to scene, 57
  - adding to stage, 54
  - building, 51–54
  - customizing by pinching, 96–97
  - formulating PV3D class, 50–51
  - geodesic, 99–100
  - rezzing to stage, 87–89
  - running applications, 57–60
  - running final base code, 54–57
  - slicing, 89–91
  - using `drawTriangles` to create burning, 606
- spherical (blimp) ellipsoids, 95–96**
- spin property, ParticleBlowUp class, 233**
- spinning out, slot car racing game, 442–444**
- splines**
- defined, 35
  - overview of, 35–36
  - parameterized, 438
- Spring Powered Camera**
- defined, 322
  - dropping for Web optimization, 391
  - instantiating for Jiglib Hello World, 351–356
  - star cruiser using, 324–326
- sprites**
- creating pool balls with bullet imprint, 380
  - saving CPU usage with point, 460
  - viewports extending, 51

**squash and stretch, animation principle, 28–39, 62**

**src folder, 34**

**stage**

- adding viewport to, 52
- depth sorting objects on, 20
- getting program from Flash library to, 113
- optimizing for web, 392
- positioning panels using XML, 470–472
- removing and adding models to, 184
- rezzing primitives to. See primitives (prims), rezzing to stage
- setting, 65–66

**stars**

- building planetarium in Flash CS4, 273–275
- building planetarium in PV3D, 269–273
- cruising using cameras, 324–326
- getting online, 268

**starter code, Wii, 393–397**

**state engines, recording 3D animations, 514–515**

**states, game**

- adding in Flash Catalyst, 588–589
- building 3D games for Wii, 367
- filling, 370
- graphics layout in Flash Catalyst, 586–587
- programming panel containers, 370–373
- using modal control, 367–368
- view states, 368–369

**Static Properties and Methods, 239–240**

**StatsView, 175–176**

**steer parameter, Wii virtual driving, 401**

**steering mechanism, 195–196**

**String Theory, 322**

**studio, PV3D, 518–519**

**styling, CNN news feed, 579–583**

**sub-objects, 35, 36**

**SubsetDataconfig.as file, Flex data wizard, 280**

**SubsetData.php file, Flex data wizard, 280**

**SubsetDataScript.as file, Flex data wizard, 280**

**Super Prim, CS4**

- creating, 120–125
- video on, 145–147

**SWF Encrypt, 422**

**swf files**

- adding animation to muzzle, 379
- adding simple preloader, 360

building viewer in Jiglib, 356–359

insecurity of, 421–422

**SWFObject search engine optimizer, 484**

**Swift3D modeling application, 179**

**switch statement, Flex, 280**

**Sync event, instantiating RSO, 425**

**synchronous file operations, AIR, 307**

## T

**tags, XML, 263–265**

**template code**

- adding materials to prim, 130–131
- prim, using wireframe material, 80–81
- shaders, 150–151

**terminology, for 3D in this book, 35–37**

**terrain editor, creating in AIR, 293–306**

- adding key listeners, 299–306
- adding sliders, 296
- drawing code, 294–295
- incorporating Adobe AIR, 306–312
- overview of, 293–294
- using buttons, 296–297
- using radio buttons, 295
- working with Perlin noise, 298–299

**terrains, modeling, 287–293**

- adding color, 289–293
- displacing vertices, 288–289
- heightmaps, 287–288
- overview of, 287

**tessellation, and cubes, 193**

**test server, installing MySQL and PHP, 276–277**

**testing, with military virtual reality, 522**

**text, website, 462**

**text nodes, XML, 264**

**TextArea component, styling CNN news feed, 579–580**

**textboxes**

- adding PONG to FLAR, 545
- Pong, 410–411

**texture**

- applying, 127–128
- low polygon models, 174
- ray gun, 377

**texture baking, 217**

**texture mapping, 134–135**

**Thales Theorem**

- applying, 5–6
- creating illusion of depth, 3
- deriving scaling equation, 6–8

**3rd person camera perspective, 323–324**  
**3D. See 3D**  
**3DSMax. See 3DSMax**  
**throttling frame animation, 52, 118**  
**Tiled property, MaterialObject3D class, 129**  
**time concept, physics, 321–322**  
**timeline animation**  
    adding to muzzle, 378–379  
    converting into ActionScript, 29–31  
**timeline scripting application, PV3D in Flash as, 41–43**  
**timer**  
    creating CS4 super prim, 121–125  
    creating game scoreboard, 383–384  
    creating loop function, 52  
**timestamps, recording 3D animations, 513–515**  
**toggling panels, 3D sites, 472–473**  
**torsion pendulum, 331**  
**Torus prim**  
    creating parametric particle system orbiting around, 17–19  
    creating tube from, 105–107  
    parametric equations for, 15–16  
    porting into PV3D from Away3D, 102–105  
**Torus worm, 17–20, 103–105**  
**tracers**  
    creating bullet, 379  
    optimizing for Web, 391–392  
**track racing. See slot car racing game**  
**trackUpdate method**  
    creating virtual highway, 441–442  
    having a wreck, 442–444  
**training, skill set, 461**  
**transformations, affine, 134–135**  
**translation, 3D, 15–21**  
    adding motion with, 60–62  
    creating parametric particle system (Torus worm), 17–20  
    creating sliding movie clip, 496–498  
    depth sorting, 20–21  
    Flash CS4 tool, 492  
    Matrix3D class handling, 4, 125–126  
    overview of, 15–16  
**transparency removal, Web optimization, 391–392**  
**transposition, in AS3, 20–21**  
**trash collection, 240–242**  
**trees**  
    converting site from 2D to 3D, 468–470  
    customizing prim, 108–110

**Triangle Culling, 121**  
**triangle-face normal vector, point lights, 148**  
**TriangleMesh3D class, vertices, 77**  
**triangles**  
    bringing down count when modeling, 173–174  
    deriving scaling equation from, 6–8  
    drawing in Flash, 128  
    drawing objects using, 127–128  
    face indices and, 77–78  
    modeling for PV3D, 173  
    putting textures on, 134  
    rendering models in 3D. *See drawTriangles class*  
**tube prim, 105–107**  
**tube radius, parametric equations, 16**  
**Tunnel class, Google map wormhole, 607–609**  
**TVStation, 418**  
**tweeting. See twitter search application**  
**twitter search application, 553–565**  
    adding interactivity in Flash Catalyst, 554–559  
    building connectivity in Flash Builder, 559–565  
    creating layout in Photoshop, 554  
    overview of, 553  
**2D. See 2D**

## U

**UAVs (unmanned aerial vehicles), 522**  
**unprojecting coordinates, Hello World, 436**  
**updates**  
    DisplayObject3D class, 160–161  
    shaders, 161–162  
**updateStage() function, parametric particle system, 17**  
**upgrades, PV3D classes, 34**  
**URLLoader class, 266**  
**Utils3D class, CS4, 114**  
**UV coordinates, 78–79**

## V

**validation, XML, 264**  
**vanishing point (asymptotic zero)**  
    adjusting to Center Stage, 12–13  
    deriving scaling equation, 7–8  
    Flash 10 automatically setting, 14  
    in Thales Theorem, 5–6  
**variable declarations, materials, 137–138**  
**Vector3D method, container-sorting, 21**  
**velocity**

- adding to `DisplayObject3D`, 326–328
  - creating constant, 408
  - particle systems and, 232
  - vertex**
    - customizing tree prim, 109
    - defined, 35
    - as element in object, 36
  - vertex normal, Gouraud Material, 154–155**
  - `Vertex3D` **class, 77**
  - vertices (verts)**
    - animation in MD2, 186–187
    - building modeling program, 286–287
    - constructing plane primitive, 78–79
    - creating cube in CS4, 114–116
    - defined, 35
    - drawing objects using materials, 127–128
    - gauging performance of, 176
    - geodesic spheres and, 99–100
    - modeling terrains by displacing, 288–289
    - as primitive element, 77
    - reducing size for Web, 391–392
    - splines composed of, 35–36
  - video, integrating particles and, 240**
  - VideoBlogger, 418**
  - `VideoStreamMaterial`, **143, 145**
  - view frustum, 38–39**
  - view states, 368–369**
  - viewable volume, view frustum, 39**
  - viewer, building JiglibFlash, 356–359**
  - viewport layer, 378**
  - Viewport3D, 54**
  - viewports**
    - building interactive cube (face level), 334–337
    - building nav system for 3D website, 474–475
    - creating CS4 super prim, 126
    - creating multiple, 53
    - defined, 39–40
    - formulating PV3D class, 51–52
    - mapping coordinates to, 39
    - size reduction for Web, 391–392
    - using `BasicView`, 67–72
  - views, FMS Administration Console, 423**
  - virtual driving, Wii, 400–401**
  - virtual highway, constructing, 441–442**
  - virtual mouse**
    - adding paddle motion, 414–415
    - adding to game, 403–404
    - integrating environments, 404
  - virtual tours, 521–551**
    - applying to military development, 521–523
    - building 3D worlds using Massive3d, 550–551
    - creating augmented reality. See augmented reality
    - `visual_scenes` library, **Collada, 195**
    - VizualPV3D, 518–519**
    - `Vnext`, **animation in MD2, 186–187**
- ## W
- Wall Carousel Course, 3D, 341–346**
    - the big problem, 344–345
    - building, 341–344
    - rapid large-scale development, 345–346
  - Walt Disney's Rules of animation, 28–29**
  - WAMP test server, installing, 276–277**
  - `weaponContainer`
    - adding animation to muzzle, 378–379
    - building ray gun, 377–378
    - game play, 390
  - weather widget, creating**
    - checking local weather, 571
    - creating code, 573–576
    - examining weather XML, 572–573
    - making widget interface, 576–577
  - Web, optimizing for, 390–392**
  - Web references**
    - Adobe Flash Collaboration Service, 458
    - Alchemy, 201
    - AS3Dmod, 360–361
    - augmented reality, 523
    - Blender, 177
    - CNN news feeds, 577
    - coding Flash and Flex, 461
    - creating action tokens, 282–283
    - CS4 Flash, 120
    - Daily Papervision, 462–463
    - DOM, 516
    - FLARToolKit, 523
    - Flash & Math, 591
    - Flash Catalyst and Flash Builder basics, 553
    - Flash Kit, 445
    - Flint particles, 249
    - FMS alternatives, 456
    - generating own markers, 532
    - getting star data from NOMAD, 268
    - Google SketchUp, 178–179
    - hooking up Web Services to Flash Builder, 583
    - Influxis account for FMS, 417
    - IPO curves, 199
    - JiglibFlash, 348

### **Web references (*continued*)**

- learning Photoshop, 461
- MD2 model packs, 181
- motion editor, 29
- Papervision3D classes, 33–34
- Papervision3D import statements, 50
- parameterized splines, 438
- parametric curves, 15
- parametric equations, 26
- particle creation, 19–20
- Pixel Bender, 207
- Pixelero blog, 606
- prims, 76
- Red 5, 457
- Runge-Kitta, 331
- Search Engine Optimization, 484
- Seb Lee-Delisle's snow systems, 244–246
- setting up Google maps, 254
- Seven Revolutions project, 462
- Spring Powered Camera, 322
- SWFObject, 485
- VizualPV3D, 518–519
- WiiFlash, 393
- WOW, 347

**Web services. See services, adding Web**

**Web spiders, 483–488**

- overview of, 483–484
- using HTML, 484
- using SWFObject, 485–488

**Webcams**

- adding video to CS4 prim from, 146–147
- for slot car racing game, 456

**Websites, developing 3D, 459–488**

- converting 2D websites. See CSIS (Center for Strategic and International Studies) website
- creating site map, 461–462
- in Flash, 460–461
- learning Photoshop, 461
- reasons for, 459–460
- Web spiders and, 483–488

**weight, animation principle, 29**

**well-formed XML documents, 263–264**

**widgets. See weather widget, creating**

**Wii, creating 3D games for, 365–416**

- adding Wii to a game, 402–404
- building car for slot car racing game, 445
- creating pool shooting game. See pool shooting game
- drawing silly string, 397–400
- Jiglib Pong. See Pong game, with Jiglib

- optimizing for Web, 390–392
- overview of, 393
- starter code, 393–397
- virtual driving, 400–401

### **wildcards, when to avoid, 115**

### **winState, 370, 588**

### **wireframe material**

- adding to prim, 132
- BitmapWireframeMaterial vs., 141–142
- prim template code using, 80–81
- properties, 129
- racking pool balls using, 381–382
- rezzing primitives to stage using. See primitives (prims), rezzing to stage
- silly string using, 397–399
- spheres. See spheres, wireframe

### **WireSphere class, 87–89**

### **world (x, y, z perspective projection) coordinates, 436–437**

### **wormhole, Google Map, 606–609**

### **Wowza, 457**

### **wrecks, in slot car racing game, 442–444**

## **X**

### **x-axis**

- 3D rotation of, 22
- building interactive particles (object level), 340–341
- constructing plane prims, 78–79
- Flash 10 clockwise rotation of, 22–23
- PV3D coordinate system and, 4
- rendering to screen and, 8–9

### **XML, 262–275**

- advantages of, 202
- binary data vs., 201
- building planetarium in Flash CS4, 273–275
- building planetarium in PV3D, 268–273
- Collada files using, 31–32
- converting site from 2D to 3D, 470–473
- feeding data into popup, 267–268
- importing into Flex and Flash, 266
- overview of, 262
- positioning panels on stage with, 470–472
- recording 3D animations, 514
- understanding, 263–264
- using E4X, 264–265

### **XML Exporter**

- 3DSMax, 204
- Blender, 202–203, 599–606

**Y****Yahoo Web services, mining, 570–577**

- checking local weather, 571
- creating code, 573–576
- examining weather XML, 572–573
- making widget interface, 576–577
- overview of, 570

**Yaw command**

- rolling wheels and moving car, 196
- rotating plane using, 81–83
- rotating sphere using, 87–88

**y-axis**

- 3D rotation of, 22
- building interactive particles (object level), 340–341
- constructing plane prim, 78–79
- Flash 10 clockwise rotation of, 22–23
- PV3D coordinate system and, 4–5
- rendering to screen and, 8–9

**Z****z component, Flash CS4, 492****z-axis**

- 3D rotation of, 22
- building interactive particles (object level), 340–341
- deriving scaling equation, 6–7
- Flash 10 clockwise rotation of, 22–23
- getting 3D with, 5
- rendering to screen and, 8–9
- in Thales Theorem, 5–6

**Z-buffer techniques, 38****zoom, camera property, 53****z-sorter property, particle systems, 232****z-sorting (depth sorting), 20–21, 23–24**