Index

A

accumulation buffer, 376, 378, 394–408
- clearing, 32, 379
- depth-of-field effect, use for, 402–406
- examples of use, 394
- full range for best results, use, 600
- motion blur, use for, 402
- sample program with depth-of-field effect, 404
- sample program with full-scene antialiasing, 397
- scene antialiasing, use for, 396

AGL, 566
- aglChoosePixelFmt(), 566, 569
- aglCopyContext(), 567, 569
- aglCreateAGLPixmap(), 567, 569
- aglCreateContext(), 567, 569
- aglDestroyAGLPixmap(), 568, 569
- aglDestroyContext(), 567, 569
- aglGetConfig(), 566, 569
- aglGetCurrentContext(), 567, 569
- aglGetCurrentDrawable(), 567, 569
- aglGetError(), 568, 569
- aglListPixelFmts(), 566, 569
- aglMakeCurrent(), 567, 569
- aglQueryVersion(), 566, 568
- aglSetOptions(), 567, 569
- aglSwapBuffers(), 568, 569
- aglUpdateCurrent(), 568, 569
- aglUseFont(), 568, 569

airbrushing, 528

Akeley, Kurt, 394

aliasing, See antialiasing

alpha, 214
- destination alpha, 236
- material properties, 197
- texture image data type, 354

alpha blending, See blending

alpha test, 384
- querying current values, 384
- rendering pipeline stage, 14, 533

ambient
- contribution to lighting equation, 207
- global light, 193, 206
- light, 173, 174, 182
- material properties, 175, 197

animation, 20–24, 600

antialiasing, 226–239
- accumulation buffer used for, 395–401
- characters (by masking), 512
- characters (by texturing), 523
- color-index mode, 232
- coverage values, 227
- enabling for points or lines, 228
- enabling for polygons, 236
- lines, 226, 228–235
- lines (by texturing), 523
- points, 228–235, 514
- polygons, 235
- RGBA mode, 229
- sample program in color-index mode, 232
- sample program in RGBA mode, 229
- sample program of filled polygons, 236
- scene, with the accumulation buffer, 396

architectural applications
- orthographic parallel projection, use of, 124

arcs, 428

aspect ratio
- perspective projection, 122
- viewport transformation, 126

atmospheric effects, See fog

attenuation of light, 183–184

attribute groups, 78–81
- client, 78
- list of, 537–559
- performance tips, 602
- server, 78
- stack depth, obtaining, 79
- stacks, 78

auxiliary buffers, 377, 380

B

back-facing polygons, 56
  culling, 57
  material property, specifying, 196
  two-sided lighting, 194
background, 29–32
  color, 29
  drawing a fixed, 382, 523
background processing, 584
backward compatibility
  tessellation, 426
  versions, 503
basis functions, 439, 440
Bernstein
  basis, 439
  polynomial, 443
Bézier
  basis, 439, 440
  curve, 443
  sample program using mesh for surface, 451
  sample program which draws curve, 441
  sample program which draws surface, 448
  surface, 446
billboarding, 219, 385
bitmaps, 278–284
  display lists cache bitmap data, 257
  distorting, 509
  drawing, 283
  feedback mode, 493
  fonts, used for, 279, 286
  imaging pipeline operations, 297
  ordering of data in, 281
  origin of, 283
  sample program, 280
  sample program that creates a font, 287
  size of, 281
bitplanes, 156, 374
  displayable colors, number of, 158
blending, 214–223, 392
  antialiasing polygons, 235
  coverage calculations for antialiasing, 227
  destination alpha, 236
  enabling, 216
  enabling for antialiasing, 229
  factors (source and destination), 215
  images, 514
  ordering polygons before drawing, 222
  rendering pipeline stage, 14, 533
  sample program for three-dimensional, 223
  sample program with blended polygons, 220
  texture function, 356
  three dimensions, in, 222
  uses of, 217
buffer, See framebuffer

C

C programming language, 8
CAD/CAM, See computer-aided design
camera analogy, 94–95
  environment mapping, 370
  viewport transformations, 125
capping, See computational solid geometry
characters
  antialiasing, 523
circles, 428
clearing the framebuffer, 29–32, 378–379
  affected by scissoring, dithering, and masking, 379, 533
  performance tips, 603
client-server, See networked operation
clip coordinates, 96, 137
  feedback mode, 493
clipping, 125
  interference regions found using clipping planes, 519
  overview, 92
  primitives in rendering pipeline, 12, 531
  viewing volume, 121
clipping planes
  additional clipping planes, 96, 136–139
  depth-buffer resolution, effect on, 600
  far, 121–125, 129
  near, 121–125, 129
  querying number of additional, 137
  sample program with additional clipping planes, 138
color
  alpha values, 214
  background, 30

cube showing blended RGB values, 155
current raster color, 285
human perception, 153
RGBA values for, 33, 154
specifying, 32
specifying for tessellation, 415
specifying in color-index mode, 164
specifying in RGBA mode, 163

color buffer, 154, 156, 374, 376, 377
clearing, 32
masking, 381

color map, 154, 159
loading for antialiasing, 232
loading for smooth shading, 167
loading, using GLUT, 583
size of, 160

color-index mode, 159–161
changing between RGBA mode and, 162
choosing between RGBA mode and, 161
coverage calculations for antialiasing, 227
dithering, 393
layering with writemasks, 381
lighting, 209–211
lighting calculations in, 210
texturing limitations, 321, 329
vertex arrays, specifying values with, 68

command syntax, 7–9

compositing images, 219

compositing transformations, 139–146

computational solid geometry, 421
capping, 390
difference of several contours, 421
interference regions, 518
intersection of two contours, 421
union of several contours, 421

Computer Graphics: Principles and Practice, xxi, 157, 593

computer-aided design
orthographic parallel projection, use of, 124

concave polygons
GLU tessellation, 410
stencil buffer, drawing with the, 516

cones, 428, 584
improving rendering of, 525

constant attenuation, 184

contours, 365

control points, 438, 442, 446, 455

convex polygons, 38

Conway, John, 526

coordinate systems
grand, fixed, 106, 115, 140
local, 106, 115, 140, 144
simple 2D, 35–36

coordinates
See clip coordinates, depth coordinates, eye coordinates, homogeneous coordinates, normalized device coordinates, object coordinates, q texture coordinates, texture coordinates, w coordinates, or window coordinates

coverage, pixel, 227

Coxeter, H. S. M., 593

cross product, 118, 589

CSG, See computational solid geometry

culling, 56–57
enabling, 57
rendering pipeline stage, 12, 532

curves and curved surfaces, 40
see also evaluators or NURBS

Curves and Surfaces for Computer-Aided Geometric Design, 439

cylinders, 428

D

data types
RGBA color conversion, 163
special OpenGL, 8
texture data, 328
warning about data type conversions, 601

decals, 385, 515
polygon offset used for, 247
texture function, 356

depth buffer, 172, 376, 377
also see hidden-surface removal
background, using masking for a common, 382
blending, use for three-dimensional, 222
clearing, 32, 172, 379
decals, for, 515

Dirichlet domains, for, 525
drawing static backgrounds, 523
masking, 381
near frustum plane effect on resolution, 600
pixel data, 295, 303

depth coordinates, 97, 128
perspective division, 128
picking use, 485
polygon offset, 247–250
rendering pipeline stage for depth-range operations, 12, 532
sample program with picking, 486
selection hit records, 474

depth test, 391
also see depth buffer
rendering pipeline stage, 14, 533

depth-cuing, See fog

depth-of-field effect, 402–406
sample program, 404

destination factor, See blending

diffuse
contribution to lighting equation, 207
light, 174, 182
material properties, 175, 197

directional light source, 182

Dirichlet domains, 524

disks, 428

display lists, 29, 253
changing mode settings, 275
compiling, 262
creating, 259
deleting, 267
disadvantages, 259, 265
editing limitations, 256
error handling, 261, 600
executing, 259, 265
executing multiple, 267
font creation, 268, 285
hierarchical, 265
immediate mode, mixing with, 265
indices for, obtaining, 262
naming, 262
nesting, 265
nesting limit, querying, 266
networked operation, 264
performance tips, 257, 602
querying use of an index, 267
rendering pipeline stage, 11
sample program creating a font, 269
sample program for creating, 253, 259
sharing among rendering contexts, 563, 575
state variables saved and restored, 274
tessellation, use with, 426
uses for, 257, 275
vertex-array data, 264
what can be stored in, 263

distorted images, 509
texture images, 359

dithering, 158–159, 392, 600
and clearing, 379
rendering pipeline stage, 14, 533

dot product
lighting calculations, use in, 207

double-buffering, 22–24
automatic glFlush(), 35
changing between single-buffering and, 162
object selection using the back buffer, 508
querying its presence, 377
sample program, 24

drawing
clearing the window, 30
forcing completion of, 33
icosahedron, 83
points, 42
polygons, 42, 55
preparing for, 29
rectangles, 40
spheres, cylinders, and disks, 428–436

drawing pixel data, See pixel data

Duff, Tom, 219

E

edge flags, 62–63
tessellated polygons generate, 414
vertex arrays, specifying values with, 68

emission, 175, 198, 206

enabling
alpha test, 384

antialiasing of points or lines, 228
antialiasing polygons, 236
blending, 216
color material properties mode, 201
culling, 57
depth test, 391
dithering, 159, 392
evaluators, 443, 447
fog, 240
lighting, 195
line stippling, 51
logical operations, 393
normal vectors for evaluated surfaces, automatic generation of, 447, 455
polygon offset, 247
polygon stippling, 58
stencil test, 386
texture coordinate generation, 369
texturing, 322, 326
unit length normal vectors ensured, 65

endianness, 300

environment mapping, 369

error handling, 501–503
error string description, 503
recommended usage, 600

evaluators, 440–454
basis functions, 439, 443
evenly spaced values, 445, 449
one-dimensional, 440
rendering pipeline stage, 11
sample program using mesh for 2D Bézier surface, 451
sample program which draws 1D Bézier curve, 441
sample program which draws 2D Bézier surface, 448
sample program which generates texture coordinates, 452
tessellation usage, 602
texture coordinates, generating, 452
two-dimensional, 446, 447

event management, using GLUT, 19

example programs, See programs

extensions
vendor-specific, 505

eye coordinates, 96, 137
texture coordinate generation, 364, 369

F

fade effect, 507

Farin, Gerald E., 439

feedback, 491–498
array contents, 497
pass-through markers, 494
querying current rendering mode, 472
returned data, 493
sample program, 495
steps to perform, 492
tessellation, obtaining vertex data after, 426

Feiner, Steven K., xxi, 593

field of view, 100
calculate, using trigonometry to, 130

filtering, 344–346
mipmapped textures, 338–344, 346
texture border colors, 361

flat shading, 165

flight simulation
fog, use of, 239

flushing, 33, 600

fog, 239–247
blending factors, 243
color-index mode, 244
density, 244
enabling, 240
equations, 243
hints, 240
RGBA mode, 244
sample program in color-index mode, 245
sample program in RGBA mode, 240

Foley, James D., xxi, 157, 593

fonts, 285–289
antialiased characters (by masking), 512
antialiased characters (by texturing), 523
bitmapped, 286
creating with display lists, 268
drawing, 284
drawing as bitmaps, 279
multi-byte, 286
same program, 287
sample program using multiple display lists, 269
X fonts, using, 564

Foran, Jim, 372

- foreshortening, perspective, 120
- fragments, 156, 374
  - alpha test, 384
  - blending, 215
  - depth test, 391
  - rendering pipeline operations, 13, 533
  - scissor test, 383
  - tests, 383–393
  - texture functions, 356
- framebuffer, 156, 375
  - capacity per pixel, 376
  - clearing, 378–379
  - copying pixel data within, 290, 295, 296
  - enabling for reading, 380
  - enabling for writing, 380
  - minimum configuration with the X Window System, 376
  - querying color resolution, 156
  - reading pixel data from, 290, 292
  - writing pixel data to, 290, 294
- front-facing polygons, 56
  - specifying material property for, 196
  - two-sided lighting, 194
- frustum, 120
- ftp (file-transfer protocol) site
  - GLUT source code, xxii
  - GLX specification, 562
  - OpenGL Programming Guide, xxii
- *Fundamentals of Computer Aided Geometric Design*, 439
- *Fundamentals of Three-Dimensional Computer Graphics*, 318

G

- Game of Life, 526
- gamma correction, 157
- Gardner, Martin, 526
- geometric primitives, 37–48, 530–532
  - performance when specifying, 603
  - rendering pipeline stage, 12
- geosciences
  - use of texturing in applications, 364
- giraffe, 160
- glAccum(), **395**
- glAlphaFunc(), **384**
- glAreTexturesResident(), **352**
- glArrayElement(), **71**
  - legal between glBegin() and glEnd(), 46
- Glassner, Andrew S., xxi
- glBegin(), 42, **43**, 414
  - restrictions, 45
- glBindTexture(), 326, **348**
- glBitmap(), 279, **283**
  - feedback mode, 493
  - fonts, used for, 286
  - imaging pipeline operations, 297
  - pixel-storage modes effect, 299
- glBlendFunc(), **216**
- glCallList(), 256, 259, **265**
  - legal between glBegin() and glEnd(), 46
- glCallLists(), **268**
  - fonts, use for, 285
  - legal between glBegin() and glEnd(), 46
  - sample program, 287
- glClear(), 30, **31**, **379**, 533
  - depth buffer, clearing the, 172
- glClearAccum(), 32, **379**
- glClearColor(), 30, **31**, **379**
- glClearDepth(), 31, **379**
- glClearIndex(), 32, **165**, **379**
  - fog, use with, 245
- glClearStencil(), 32, **379**
- glClipPlane(), **137**
- glColor*(), 33, **163**
  - legal between glBegin() and glEnd(), 46
- glColorMask(), 379, **381**
- glColorMaterial(), **201**
  - performance tips, 602
- glColorPointer(), **68**
- glCopyPixels(), 290, **295**
  - alternative uses, 527
  - dithering, turn off, 600
  - feedback mode, 493
  - glReadBuffer() effect, 380
  - imaging pipeline operations, 296
  - pixel-transfer modes effect, 302

glCopyTexImage1D(), **336**
  glReadBuffer() effect, 380
  pixel-transfer modes effect, 302
glCopyTexImage2D(), **329**
  glReadBuffer() effect, 380
  pixel-transfer modes effect, 302
glCopyTexSubImage1D(), **337**, 337
  glReadBuffer() effect, 380
  pixel-transfer modes effect, 302
glCopyTexSubImage2D(), **335**
  glReadBuffer() effect, 380
  pixel-transfer modes effect, 302
glCullFace(), **57**
glDeleteLists(), **267**, 286
glDeleteTextures(), **351**
glDepthFunc(), **391**
glDepthMask(), **381**
  blending opaque and translucent objects, 223
glDepthRange(), **128**
  gluUnProject(), relationship to, 147
glDisable(), 10, **48**
glDisableClientState(), **68**
glDrawArrays(), **74**
glDrawBuffer(), 295, **380**
glDrawElements(), **72**
glDrawPixels(), 290, **294**, 387, 524
  alternative uses, 527
  feedback mode, 493
  pixel-storage modes effect, 299
  pixel-transfer modes effect, 302
glEdgeFlag*(), **63**
  legal between glBegin() and glEnd(), 46
glEdgeFlagPointer(), **68**
glEnable(), **48**, 178
  also see enabling
glEnableClientState(), 46, **67**
glEnd(), 42, **43**, 414
  restrictions, 45
glEndList(), 256, 259, **263**
glEvalCoord*(), **445**, **447**
  legal between glBegin() and glEnd(), 46
  used instead of glVertex*(), 440, 443
glEvalMesh*(), **445**, **449**
glEvalPoint*()
  legal between glBegin() and glEnd(), 46
glFeedbackBuffer(), **492**
  glRenderMode(), use with, 472
glFinish(), **35**
glFlush(), 34, **35**, 600
glFog*(), **243**
glFrontFace(), **56**
glFrustum(), 101, 121, **121**, 533
glGenLists(), 256, **262**
  fonts, use for, 286
glGenTextures(), 326, **347**
glGetBooleanv(), 10, **49**, 537
  double-buffering support, querying, 377
  stereo support, querying, 377
glGetClipPlane(), 536
glGetDoublev(), 10, **49**, 537
glGetError(), 10, **502**, 536
glGetFloatv(), 10, **49**, 537
  line width attributes, obtaining, 51
  point size attributes, obtaining, 50
glGetIntegerv(), 10, **49**, 537
  alpha test information, obtaining, 384
  attribute stack depth, obtaining, 79
  clipping planes, obtaining number of additional, 137
  color resolution, obtaining, 156
  display list nesting limit, obtaining, 266
  matrix stack depth, obtaining, 135
  maximum texture size, obtaining, 330
  name stack depth, obtaining, 473
  pixel map information, obtaining, 304
  rendering mode, obtaining current, 472
  stencil-related values, obtaining, 386
glGetLight*(), 10, 536
glGetMap*(), 536
glGetMaterial*(), 536
glGetPixelMap*(), 536
glGetPointerv(), 10, **49**, 537
glGetPolygonStipple(), 10, 536
glGetString(), **503**, 536
glGetTexEnv*(), 536
glGetTexGen*(), 536
glGetTexImage(), 536

pixel-storage modes effect, 299
pixel-transfer modes effect, 302
glGetTexLevelParameter*(), **331**, 536
glGetTexParameter*(), 536
texture residency, obtaining, 351
glHint(), **228**
fog use, 240
texture use, 326
glIndex*(), **164**
fog, use with, 245
legal between glBegin() and glEnd(), 46
glIndexMask(), 379, **381**
glIndexPointer(), **68**
glInitNames(), 471, 472, **473**
glInterleavedArrays(), **76**
glIsEnabled(), 10, **48**, 537
glIsList(), **267**
glIsTexture(), **347**
glLight*(), 178, **180**, 181, 186
glLightModel*(), **193**
glLineStipple(), **51**
glLineWidth(), **50**
glListBase(), **267**
fonts, use for, 286
sample program, 287
glLoadIdentity(), 101, **103**, 112, 533
performance tips, 602
viewing transformations, use before, 99
glLoadMatrix*(), 102, **104**, 104, 533
glLoadName(), 472, **474**
glLogicOp(), **394**
glMap*(), 442, **443**, **446**
glMapGrid*(), **445**, **449**
glMaterial*(), 179, **196**
legal between glBegin() and glEnd(), 46
performance tips, 602
glMatrixMode(), 101, **103**
use with matrix stacks, 133
glMultMatrix*(), 102, **104**, 533
performance tips, 602
glNewList(), 256, 259, **262**
glNormal*(), **64**
legal between glBegin() and glEnd(), 46
glNormalPointer(), **68**
glOrtho(), **124**, 533
picking matrix use, 479
glPassThrough(), 492, **494**
glPixelMap*(), **304**
glPixelStore*(), **299**
cannot be stored in display lists, 264
polygon stippling, 58
texture image data, effect on, 328, 329, 332, 335, 336
glPixelTransfer*(), **302**, 524
texture image data, effect on, 328, 329, 332, 335, 336
glPixelZoom(), **305**, 509
glPointSize(), **50**
glPolygonMode(), **56**
antialiasing, effect on, 235
polygon offset, use with, 247
glPolygonOffset(), **248**
glPolygonStipple(), **58**
pixel-storage modes effect, 299
glPopAttrib(), 10, **79**, 274, 537
glPopClientAttrib(), 10, **81**, 537
glPopMatrix(), **133**, 143, 189, 274
restore orientation of coordinate systems, 146
selection, use with, 471
glPopName(), 472, **473**
glPrioritizeTextures(), **353**
glPushAttrib(), 10, **79**, 274, 537
glPushClientAttrib(), 10, **81**, 537
glPushMatrix(), **133**, 143, 189, 274
save orientation of coordinate systems, 146
selection, use with, 471
glPushName(), 471, 472, **473**
glRasterPos*(), 279, **282**
images, for positioning, 290
selection hits, can cause, 474
glReadBuffer(), 295, **380**
glReadPixels(), 290, **292**
glReadBuffer() effect, 380
pixel-storage modes effect, 299
pixel-transfer modes effect, 302
glRect*(), **40**
glRenderMode(), 471, **472**, 474, 492

- glRotate*(), **109**, 140, 143, 533
  - performance tips, 602
- glScale*(), 99, **110**, 143, 533
  - performance tips, 602
- glScissor(), **383**
- glSelectBuffer(), 471, **472**
  - display lists, cannot be stored in, 264
- glShadeModel(), **165**
- glStencilFunc(), **385**
- glStencilMask(), **381**
- glStencilOp(), **386**
- glTexCoord*(), 326, **358**
  - legal between glBegin() and glEnd(), 46
- glTexCoordPointer(), **68**
- glTexEnv*(), 326, **354**
- glTexGen*(), **364**
  - environment mapping, 370
- glTexImage1D(), **335**
  - pixel-storage modes effect, 299
  - pixel-transfer modes effect, 302
- glTexImage2D(), 326, **327**
  - pixel-storage modes effect, 299
  - pixel-transfer modes effect, 302
  - specifying mipmaps, 339
- glTexParameter*(), 326, **363**
  - specifying filtering methods, 345
- glTexSubImage1D(), **336**
  - pixel-storage modes effect, 299
  - pixel-transfer modes effect, 302
- glTexSubImage2D(), **332**
  - pixel-storage modes effect, 299
  - pixel-transfer modes effect, 302
- glTranslate*(), **108**, 140, 143, 533
  - performance tips, 602
- GLU, 2, 14, 410
  - drawing spheres, cylinders, and disks, 428–436
  - error string description, 503
  - obsolete routines
    - gluBeginPolygon(), 427
    - gluEndPolygon(), 427
    - gluNextContour(), 427
  - quadrics, 428–436
  - tessellation, 39, 410–428
  - version numbers, obtaining, 504
- gluBeginCurve(), 455, **464**
- gluBeginSurface(), 455, **463**
- gluBeginTrim(), **465**
- gluCylinder(), 429, **431**
- gluDeleteNurbsRenderer(), **460**
- gluDeleteQuadric(), 429, **429**
- gluDeleteTess(), **426**, 427
- gluDisk(), 429, **432**
- gluEndCurve(), 455, **464**
- gluEndSurface(), 455, **463**
- gluEndTrim(), **465**
- gluErrorString(), 429, 462, **503**
  - polygon tessellation, 414
- gluGetNurbsProperty(), **461**, 536
- gluGetString(), **505**, 536
- gluGetTessProperty(), **422**, 536
- gluLoadSamplingMatrices(), **461**
- gluLookAt(), 97, 99, **116**, 140
- gluNewNurbsRenderer(), 455, **459**
- gluNewQuadric(), 428, **429**
- gluNewTess(), **412**, 427
- gluNurbsCallback(), 455, **462**
- gluNurbsCurve(), 455, **464**
- gluNurbsProperty(), 455, **460**
- gluNurbsSurface(), 455, **463**
- gluOrtho2D(), **125**, 601
  - resized windows, use with, 36
- gluPartialDisk(), 429, **432**
- gluPerspective(), 101, **123**, 140
  - picking matrix use, 479
- gluPickMatrix(), **479**
- gluProject(), **150**
- gluPwlCurve(), **465**
- gluQuadricCallback(), 429, **429**
- gluQuadricDrawStyle(), 428, **430**
- gluQuadricNormals(), 428, **430**
- gluQuadricOrientation(), 428, **430**
- gluQuadricTexture(), 429, **431**
- gluScaleImage(), **329**
- gluSphere(), 429, **431**

GLUT, 15, 579–585
basic functions, 16–20
event management, 19
glutCreateWindow(), 17, **581**
glutDisplayFunc(), 17, **581**
glutIdleFunc(), 20, **584**
glutInit(), 16, **580**
glutInitDisplayMode(), 16, **580**
glutInitWindowPosition(), 17, **581**
glutInitWindowSize(), 17, **581**
glutKeyboardFunc(), 19, **582**
glutMainLoop(), 17, **585**
glutMotionFunc(), 19, **582**
glutMouseFunc(), 19, **582**
glutPostRedisplay(), 17, 256, **582**
glutReshapeFunc(), 19, **582**
simple example, 35
glutSetColor(), 16, 165, 210, **583**
smooth shading, use for, 167
glutSolidCone(), **584**
glutSolidCube(), 20, **583**
glutSolidDodecahedron(), **584**
glutSolidIcosahedron(), **584**
glutSolidOctahedron(), **584**
glutSolidSphere(), 20, **583**
glutSolidTeapot(), **584**
glutSolidTetrahedron(), **584**
glutSolidTorus(), **583**
glutSwapBuffers(), 23
glutWireCone(), **584**
glutWireCube(), 20, **583**
glutWireDodecahedron(), **584**
glutWireIcosahedron(), **584**
glutWireOctahedron(), **584**
glutWireSphere(), 20, 140, **583**
glutWireTeapot(), **584**
glutWireTetrahedron(), **584**
glutWireTorus(), **583**
sample program introducing GLUT, 18
window management, 16, 35

gluTessBeginContour(), **423**

gluTessBeginPolygon(), **423**

gluTessCallback(), **412**, 423, 427

gluTessEndContour(), **423**

gluTessEndPolygon(), **423**

gluTessNormal(), 422, **422**, 426

gluTessProperty(), **417**, 423

gluTessVertex(), **423**, 427

gluUnProject(), **147**, 150

glVertex*(), **41**
legal between glBegin() and glEnd(), 46
using glEvalCoord*() instead, 440

glVertexPointer(), 46, **68**

glViewport(), 102, **126**
using with resized windows, 36

GLX, 14, 562
ftp site for GLX specification, 562
glXChooseVisual(), 563, 604
glXCopyContext(), 563
glXCreateContext(), 563
glXCreateGLXPixmap(), 563
glXDestroyContext(), 563
glXDestroyGLXPixmap(), 563
glXGetClientString(), 562
glXGetConfig(), 376, 563
glXGetCurrentContext(), 563
glXGetCurrentDisplay(), 563
glXGetCurrentDrawable(), 563
glXIsDirect(), 563
glXMakeCurrent(), 563
glXQueryExtension(), 562
glXQueryExtensionsString(), 562
glXQueryServerString(), 562
glXQueryVersion(), 562
glXSwapBuffers(), 23, 564
glXUseXFont(), 564
glXWaitGL(), 564
performance tips, 603

glXWaitX(), 564
performance tips, 603
Gouraud shading, See smooth shading

H

Haeberli, Paul, 372, 394
haze, See fog
header file, 15
hidden-line removal, 521
polygon offset used for, 247
hidden-surface removal, 171–173, 391
hierarchical models, 132, 265
picking, 483–485
highlights, See specular
hints, 228
fog, 240
perspective correction, 228, 326
hits (selection), See selection (hit records)
holes in polygons, 38, 518
homogeneous coordinates, 37, 594
Hoschek, Josef, 439
Hughes, John F., xxi, 593

I

IBM OS/2 Presentation Manager to OpenGL Interface, see PGL
icosahedron, drawing, 83
identity matrix, 99, 103, 112, 602
illumination, See lighting
images, 278, 289–295
also see pixel data
blending, 514
compositing, 215
distorted, 509
imaging pipeline, 291, 296–308
interpolating between, 514
magnifying or reducing, 305
nonrectangular, 219
projecting, 523
sample code which draws an image, 294
sample program which draws, copies, and zooms an image, 306
scaling and rotating, 523
sources of, 290
superimposing, 515
transposing, 528
warping, 523
imaging pipeline, See images (imaging pipeline)
immediate mode, 29, 252
display lists, mixing with, 265
infinite light source, 182
input events
handling, using GLUT, 19
intensity
texture image data type, 354
Interactive Inspection of Solids: Cross-sections and Interferences, 518
interference regions, 518
interleaved arrays, 75
interpolating
color values and texture coordinates, 228, 357
invariance
of an OpenGL implementation, 600, 605
Inventor, see Open Inventor

J

jaggies, 226
jittering, 396, 401, 407
accFrustum() routine, 396
accPerspective() routine, 396
sample code to jitter projection transformations, 397
sample program with orthographic projection, 401

K

Kilgard, Mark, xxii, 15, 562, 579
Korobkin, Carl, 372

L

Lasser, Dieter, 439

layers, drawing, 511

Life, Game of, 526

light sources, 180–192
- ambient light, 174, 182
- contribution to lighting equation, 206
- diffuse light, 174, 182
- directional, 182
- display lists cache values, 258
- infinite light source, 182
- local light source, 182
- maximum number of sources, 178
- moving along with the viewpoint, 191
- moving light sources, 187–192
- multiple light sources, 186
- performance tips, 178
- positional, 182
- rendering pipeline stage, 12, 531
- RGBA values, 175
- sample program that moves the light source, 189
- specifying a light source, 178
- specular light, 174
- spotlights, 184–186
- stationary, 187

lighting
- also see light sources, material properties
- ambient light, 173
- approximation of the real world, 173
- attenuation, 183–184
- calculations in color-index mode, 210
- color-index mode, 209–211
- default values, using, 179
- display lists cache values, 258
- enabling, 178, 179
- enabling and disabling, 195
- equation that calculates lighting, 205
- global ambient light, 193, 206
- lighting model, 192–195
- lighting model, specifying a, 179
- rendering pipeline stage, 12, 531
- sample program introducing lighting, 176
- steps to perform, 176
- two-sided materials, 194
- viewer, local or infinite, 194

line segment, 38

linear attenuation, 184

lines, 38
- antialiasing, 228–235, 523
- connected closed loop, specifying, 43, 44
- connected strip, specifying, 43, 44
- feedback mode, 493
- querying line width, 51
- sample program with wide, stippled lines, 53
- specifying, 43, 44
- stippling, 51
- tessellated polygons decomposed into, 414
- width, 50

local light source, 182

logical operations
- rendering pipeline stage, 14, 533
- transposing images, using for, 528

lookup table, See color map

luminance, 292, 314
- pixel data formats for, 293, 298
- texture image data type, 354

M

magnifying images, 305

masking, 381
- antialiasing characters, 513
- layers, drawing, 511
- rendering pipeline stage, 14, 533

material properties, 179, 195–204
- ambient, 175, 197
- changing a single parameter with glColorMaterial(), 201
- changing material properties, 199
- diffuse, 175, 197
- display lists cache values, 258
- emission, 175, 198, 206
- enabling color material properties mode, 201
- performance when changing, 602
- rendering pipeline stage, 12, 531
- RGBA values, 176

sample program which changes material properties, 199
sample program which uses glColorMaterial(), 202
shininess, 198
specular, 175, 198
two-sided lighting, 194

matrix
also see matrix stack
choosing which matrix is current, 103
column-major ordering, 104
current, 99
danger of extensive changes, 600
display lists cache matrix operations, 257
identity, 99, 103, 112, 602
loading, 104
modelview, 96, 103
multiplying matrices, 104
NURBS, specifying for sampling, 461
orthographic parallel projection, 598
perspective projection, 598
projection, 101, 103
rotation, 596
row-major ordering, 104
scaling, 596
texture, 371
transformation pipeline, 96
transformations of homogeneous coordinates, 594
translation, 596

matrix stack, 132–136
choosing which matrix stack is current, 133
current matrix stack, 533
modelview, 135
popping, 133
projection, 135
pushing, 133
querying stack depth, 135
texture, 371

Megahed, Abe, 518

Microsoft
Microsoft Win32, See Win32
Microsoft Windows, 14
Microsoft Windows 95, 574
Microsoft Windows NT, xxii
Microsoft Windows to OpenGL interface, See WGL

mipmapping, 338–344
minification filters, 346
texture objects for mipmaps, 350

mirroring objects, See scaling

modeling transformations, 99, 104, 108–113
camera analogy, 94
connection to viewing transformations, 100
example, 111
rotation, 109
rotation matrix, 596
sample program, 112
scaling, 110
scaling matrix, 596
translation, 108
translation matrix, 596

models
rendering wireframe and solid, 20, 583

modelview matrix, 96, 103
arbitrary clipping planes, effect on, 137
stack, 135

motion blur, 402
stippling, with, 507

motion, See animation

movie clips, 527

multiple layers
displaying with overlap, 511

N

name stack, 471–475
creating, 472
initializing, 472
loading, 472
multiple names, 483–485
popping, 472
pushing, 472
querying maximum depth, 473

networked operation, 34–35
attribute groups, saving and restoring, 78
display lists, 264
versions, 504

Non-Uniform Rational B-Splines, see NURBS

nonplanar polygons, 39

normal vectors, 63–65, 178
- calculating, 588
- calculating for analytic surfaces, 588
- calculating for polygonal data, 591
- calculating length, 65
- cross product, calculating normalized, 85
- enabling automatic unit length division, 65
- inverse matrix generated, 533
- matrix transformations, 96
- normalized, 65
- NURBS, generating for, 463
- quadrics, generated for, 430
- rendering pipeline stage, 12, 531
- specifying, 64
- tessellation, specifying for, 415
- transformations, 595
- unit length optimizes performance, 603
- vertex arrays, specifying values with, 68

normal, See normal vectors

normalized device coordinates, 96

NURB Curves and Surfaces (book title), 439

NURBS, 455–468
- creating a NURBS curve or surface, 462–464
- creating a NURBS object, 459
- culling, 460
- deleting a NURBS object, 460
- display list use, 256
- error handling, 462
- method of display (lines or filled polygons), 460
- normal vectors, generating, 463
- properties, controlling NURBS, 460
- querying property value, 461
- references, 439
- sample program which draws a lit NURBS surface, 456
- sample program with a trimmed surface, 467
- sampling precision, 460
- source for matrices, 461
- steps to use, 455
- texture coordinate generation, 463
- trimming, 464–468

NURBS Book, The, 439

NURBS for Curve and Surface Design, 439

O

object coordinates, 96
- texture coordinate generation, 364

objects, See models

opacity, 215

Open Inventor, 3, 15

OpenGL Extension to the X Window System, see GLX

OpenGL Programming for the X Window System, xxii, 15, 16, 562, 579

OpenGL Reference Manual, xxi, 529, 536, 562

OpenGL Utility Library, see GLU

OpenGL Utility Toolkit, see GLUT

orthographic parallel projection, 101, 124–125
- jittering, 400
- matrix, 598
- specifying with integer coordinates, 601

outlined polygons, 55, 63
- polygon offset solution, 247

overlapping objects, 518

P

painting, 215, 218, 528

partial disks, 428

pass-through markers, 494

performance tips
- clearing the window, 32
- display lists, 256, 257
- flat shading, 603
- flushing the pipeline, 34
- fog, 240
- GLX tips, 603
- hints, 228
- light source attenuation, effect of, 184
- light sources, effect of additional, 178
- list of general tips, 602
- material properties, changing, 602
- NURBS and display lists, 256
- pixel data alignment, 301
- pixel data, drawing, 314
- polygon restrictions, 39
- polygon subdivision, 82

pushing and popping attribute groups, 602
rasterization and fragment operations for pixel data, 603
removing hidden surfaces, 173
specifying geometric primitives, 603
tessellation and display lists, 256
tessellation, use of, 426
texture images, internal format of, 328
texture objects, 346, 602
texture subimages, 602
two-sided lighting, 195
unit-length normal vectors, 603
vector and scalar forms of commands, 603
vertex arrays, 603

perspective projection, 120–123
correction hint, 228, 326
depth coordinates, effect on, 128
jittering, 397
matrix, 598
perspective division, 96

PGL, 14
pglChooseConfig(), 570, 572
pglCopyContext(), 571, 572
pglCreateContext(), 571, 572
pglDestroyContext(), 571, 572
pglGetCurrentContext(), 571, 573
pglGetCurrentWindow(), 571, 573
pglGrabFrontBitmap(), 571, 573
pglIsIndirect(), 571, 572
pglMakeCurrent(), 571, 572
pglQueryCapability(), 570, 572
pglQueryConfigs(), 570, 572
pglQueryVersion(), 570, 572
pglReleaseFrontBitmap(), 571, 573
pglSelectColorIndexPalette(), 572, 573
pglSwapBuffers(), 571, 573
pglUseFont(), 572, 573
pglWaitGL(), 571, 573
pglWaitPM(), 571, 573

picking, 478–488
back buffer for, using the, 508
depth coordinates, 485
hierarchical models, 483–485
projection matrix, special, 479
sample program, 480
sample program with depth coordinates, 486
strategies, 489
sweep selection, 490

Piegl, Les, 439

pipeline
geometric processing, 530–532
imaging, 291, 296–308
rendering, 10–14
vertex transformation, 96

pixel
coverage, 227

pixel data, 278, 289–295
also see images
byte alignment, 301
byte ordering, 300
copying within the framebuffer, 13, 290, 295, 296, 532
depth buffer pixel data, 295, 303
drawing or reading a subrectangle of, 300
drawing process in detail, 309–312
endianness, 300
feedback mode, 493
formats for reading or drawing, 292
formats for storing in memory, 293, 298
mapping, 12, 304, 532
packing into processor memory, 13, 298–300, 532
performance tips, 314
pipeline operations, 12, 291, 296–308, 532
pixel zoom, 305
querying pixel mapping information, 304
reading from the framebuffer, 290, 292
reading process in detail, 312–314
sample code which draws an image, 294
sample program which draws, copies, and zooms pixel data, 306
stencil buffer pixel data, 293, 303
storage modes, 299
transfer modes, 13, 302, 354, 532
unpacking from processor memory, 12, 298–300, 532
writing to the framebuffer, 290, 294

point light source, See positional light source

points, 37
antialiasing, 228–235, 514
drawing, 42
feedback mode, 493
querying point size, 50
round, 228–235, 514
size, 49

specifying, 43, 44

polygon offset, 247–250
depth slope of a polygon, 248
enabling, 247
hidden-line removal, 521

polygonal approximations to surfaces, 81

polygons, 38
boundary edges, 62–63
concave, drawing filled, 410, 516
convex, 38
culling the faces, 56
drawing, 42
drawing as points, lines, or filled, 55
feedback mode, 493
front and back faces, 56
holes in, 38
non-convex, 38, 62
nonplanar, 39
polygon mode, 12, 55, 532, 603
reversing the faces, 56
sample program with stippled polygons, 60
self-intersecting, 416
simple, 38
specifying, 43, 45
stippling, 57
tessellation, specifying for, 423
Voronoi, 524

positional light source, 182

primitives
geometric, 37–48
raster, 278

priority of texture objects, 352

Procedural Elements for Computer Graphics, 428

programs
aaindex.c, 232
aapoly.c, 236
aargb.c, 229
accanti.c, 401
accpersp.c, 397
alpha3D.c, 223
alpha.c, 220
bezcurve.c, 441
bezmesh.c, 451
bezsurf.c, 448
checker.c, 323
clip.c, 138
colormat.c, 202
cube.c, 98
dof.c, 404
double.c, 24
drawf.c, 280
feedback.c, 495
fog.c, 240
fogindex.c, 245
font.c, 287
ftp site, xxii
hello.c, 18
image.c, 306
light.c, 176
lines.c, 53
list.c, 259
material.c, 199
mipmap.c, 340
model.c, 112
movelight.c, 189
pickdepth.c, 486
picksquare.c, 480
planet.c, 141
polys.c, 60
quadric.c, 433
robot.c, 144
select.c, 475
smooth.c, 166
stencil.c, 387
stroke.c, 269
surface.c, 456
tess.c, 414, 415, 424
texbind.c, 348
texgen.c, 365
texsub.c, 333
texturesurf.c, 452
torus.c, using a display list, 253
trim.c, 467
unproject.c, 148
varray.c, 69

projecting images, 523

projection matrix, 101, 103
matrix stack, 135
orthographic parallel projection matrix, 598
perspective projection matrix, 598
shadows created with, 520

projection transformations, 100, 120–125
camera lens analogy, 94

collapsing geometry to a single plane, 600
jittering, 397, 400
orthographic parallel, 101, 124–125, 601
perspective, 120–123
picking, 479
texturing effects, 372
two-dimensional, 125

proxy textures, 330

Q

q texture coordinates, 372
avoiding negative values, 601

quadratic attenuation, 184

quadrics, 428–436
creating an object, 429
destroying an object, 429
drawing as points, lines, and filled polygons, 430
error handling, 429
normal vectors, generating, 430
orientation, 430
quadratic equation, 428
sample program, 433
steps to use, 428
texture coordinates, generating, 431

quadrilateral
specifying, 43, 45
strip, specifying, 43, 45

R

raster position, 282
after drawing a bitmap, 283
current, 282, 533
current raster color, 285
current, obtaining the, 282
transformation of, 282

rasterization, 156, 374
exact, two-dimensional, 601
rendering pipeline stage, 13

reading pixel data, See pixel data

Real Projective Plane, The, 593

rectangles
specifying, 40

reducing images, 305

reflecting objects, See scaling

reflection, See material properties

reflective objects, See environment mapping

refresh, screen, 21

removing hidden surfaces, See hidden-surface removal

repeatability, 606

resident textures, 332, 351
management strategies, 352
querying residence status, 351

RGBA mode, 157
changing between color-index mode and, 162
choosing between color-index mode and, 161
coverage calculations for antialiasing, 227
data type conversion, 163
light source colors, 175
lighting calculations in, 205
material property values, 176
vertex arrays, specifying values with, 68

robot arm example, 143–146

Rogers, David, 428

Rossignac, Jarek, 518

rotating images, 523

rotation, 109
matrix, 596

S

sample programs, See programs

scaling, 110
matrix, 596

scaling images, 523

Schneider, Bengt-Olaf, 518

Scientific American, 526

scissor test, 383
and clearing, 379
rendering pipeline stage, 14, 533

Segal, Mark, 372

selection, 470–491
back buffer for, using the, 508
hit records, 474

programming tips, 489
querying current rendering mode, 472
rendering pipeline stage, 532
sample program, 475
steps to perform, 471
sweep selection, 490
shading
flat, 165
performance tips, 603
sample program with smooth shading, 166
smooth, 165
specifying shading model, 165
shadows, 205, 406, 520
shininess, 198
also see environment mapping
silhouette edges, 82
smoke, See fog
smooth shading, 165
solar system example, 140–143
source factor, See blending
specular
contribution to lighting equation, 208
light, 174
material properties, 175, 198
spheres, 428, 583
split-screen
multiple viewports, 126
spotlights, See light sources
state machine, 9–10
state variables, 48
attribute groups, 78–81
display list execution, effect of, 273
enable and disable states, 48
list of, 537–559
performance of storing and restoring, 602
querying, 49
stencil buffer, 376, 377
clearing, 32, 379
concave polygons, for drawing, 516
decals, for, 515
Dirichlet domains, for, 525
Game of Life, for the, 526
hidden-line removal, 522
masking, 381
pixel data, 293, 303
stencil test, 385–391
examples of using, 387
interference regions found using clipping planes, 519
querying stencil parameters, 386
rendering pipeline stage, 14, 533
sample program, 387
stereo, 377, 380
querying its presence, 377
stippling
display lists cache stipple patterns, 258
enabling line stippling, 51
enabling polygon stippling, 58
fade effect, use for, 507
line pattern reset, 52, 493, 497
lines, 51
polygons, 57
sample program with line stipple, 53
sample program with polygon stippling, 60
stencil test, use of, 391
translucency, use to simulate, 506
stitching, 247
stretching objects, See scaling
stride
vertex arrays, 70, 76
subdivision, 81–89
generalized, 88
icosahedron example, 86
recursive, 88
subimages, 332–335, 336
superimposing images, 515
surface normals, See normal vectors
surfaces, See evaluators or NURBS
swapping buffers, See double-buffering
syntax, See command syntax

T

Terminator 2, 370
tessellation, 39, 410–428
backward compatibility with obsolete routines, 426
begin and end callback routines, 414
callback routines, 412–417

combine callback routine, 414, 416
computational solid geometry, winding rules used for, 421
contours, specifying, 423
converting code to use the GLU 1.2 tessellator, 427
creating an object, 412
decomposition into geometric primitives, 414
deleting objects, 426
display list use, 256
edge flag generation, 414
error handling, 414
evaluators used to perform, 602
interior and exterior, determining, 418–422
intersecting contours combined, 414, 416
performance tips, 426
polygons, specifying, 423
properties, 417–422
reuse of objects, 412, 426
reversing winding direction, 422
sample code, 414, 415, 424
user-specified data, 417
vertices, specifying, 415, 423
winding rules, 418–422

texels, 14, 318

text, see characters

texture coordinates, 326, 357–371
assigning manually, 357
avoiding negative q values, 601
clamping, 360–363
computing manually, 358
enabling automatic generation of, 369
environment mapping, automatic generation for, 370
evaluators, generated by, 452
generating automatically, 364–371
NURBS, generating for, 463
q coordinate, 372
quadrics, generated for, 431
reference planes, specifying, 364
rendering pipeline stage, 12, 531
repeating, 360–363
sample program with texture coordinate generation, 365
tessellation, specifying for, 415
vertex arrays, specifying values with, 68
wrapping modes, 360–363

texture functions, 354–357
blend, 356
blending color, 356
decal, 326, 356
fragment operations, 356
modulate, 356
pixel-transfer modes effect, 354
replace, 356
texture internal format, interaction with, 354

texture images
alpha data, 354
borders, 337, 361
components, 327
data types, 328
distorting, 359
framebuffer as a source of, 329, 334, 336
imaging pipeline operations, 297
intensity data, 354
internal format, 327
luminance data, 354
mipmaps, 338–344
one-dimensional, 335–337
performance affected by internal format, 328
performance of texture subimages, 602
power of 2 size restriction, 329
proxy textures, 330
querying maximum size, 330
residence status, 351
resident textures, 332, 351
resident textures, management strategies of, 352
sample program with mipmaps, 340
sample program with subimages, 333
specifying, 326–338
subimages, 332–335, 336
working set of textures, 332, 346, 351

texture mapping, see texturing

texture matrix, 371
rendering pipeline stage, 531

texture objects, 326, 346–351
binding, 348
creating, 348
data which can be stored in, 348
deleting, 351
fragmentation of texture memory, 353
least-recently used (LRU) strategy, 353
mipmaps, 350
naming, 347

- performance tips, 346, 602
- priority, 352
- rendering pipeline, 13, 532
- sample program, 323
- sample program with multiple texture objects, 348
- sharing among rendering contexts, 563, 575
- steps to perform, 346
- using, 348

texturing
- also see texture coordinates, texture functions, texture images, texture matrix, and texture objects
- antialiasing characters, 523
- antialiasing lines, 523
- blending, 219
- border colors, treatment of, 361
- color-index mode limitations, 321, 329
- creating contours, 365
- decals with alpha testing, 385
- display lists cache texture data, 258
- enabling, 322, 326
- environment mapping, 369
- filtering, 344–346
- image transformations, 523
- mipmapping, 338–344, 346
- perspective correction hint, 326
- rendering pipeline stage, 13, 532
- sample program, 323
- sample program with evaluated, Bézier surface, 452
- sample program with mipmapping, 340
- sample program with texture coordinate generation, 365
- sample uses for, 523
- simulating shadows or spotlights, 372
- steps to perform, 321
- what's new in release 1.1, 321

3D Computer Graphics: A User's Guide for Artists and Designers, xxi

3D models, rendering, 20, 583

Tiller, Wayne, 439

tips, programming, 599
- also see performance tips
- error handling, 600
- selection and picking, 489
- transformations, 129

transformations
- also see modeling transformations, projection transformations, viewing transformations, and viewport transformations
- combining multiple, 139–146
- display lists cache transformations, 257
- general-purpose commands, 102
- matrices, 595–598
- mimicking the geometric processing pipeline, 149
- modeling, 104, 108–113
- ordering correctly, 105–108
- overview, 92
- performance tips, 602
- projection, 100, 120–125
- reversing the geometric processing pipeline, 147
- sample program, 98
- sample program combining modeling transformations, 141, 144
- sample program for modeling transformations, 112
- sample program showing reversal of transformation pipeline, 148
- troubleshooting, 129–131
- units, 123
- viewing, 104, 113–118
- viewport, 102, 125–128

translation, 108
- matrix, 596

translucent objects, 215, 506
- stencil test, creating with the, 390

transparent objects, 215
- creating with the alpha test, 385

transposing images, 528

triangle
- fan, specifying, 43, 45
- specifying, 43, 45
- strip, specifying, 43, 45
- tessellated polygons decomposed into, 414

trimming
- curves and curved surfaces, 464–468
- sample program, 467

two-sided lighting, 194

U

up-vector, 99

Utility Library, OpenGL, see GLU

Utility Toolkit, OpenGL, see GLUT

V

van Dam, Andries, xxi, 157, 593
van Widenfelt, Rolf, 372
vendor-specific extensions, 505
versions, 503–505
  GLU, 504
vertex, 37
  also see vertex arrays
  evaluators, generating with, 440
  feedback mode, 493
  per-vertex operations pipeline stage, 12, 531
  specifying, 41
  tessellation, specifying for, 415, 423
  transformation pipeline, 96
vertex arrays, 65–77
  dereference a list of array elements, 72
  dereference a sequence of array elements, 74
  dereference a single element, 71
  disabling, 68
  display list use, 264
  enabling, 67
  interleaved arrays, 75
  interleaved arrays, specifying, 76
  performance tips, 603
  querying, 537
  reuse of vertices, 74
  sample program, 69
  specifying data, 68
  steps to use, 66
  stride between data, 70, 76
video
  fake, 527
  flipping an image with glPixelZoom(), 306
  textured images, 332
viewing
  camera analogy, 94–95
viewing transformations, 99, 104, 113–118
  connection to modeling transformations, 100
  default position, 99
  different methods, 118
  pilot view, 119
  polar view, 119
  tripod analogy, 94
  up-vector, 99
viewing volume, 121
  clipping, 125, 136
  jittering, 397, 400
viewpoint
  lighting, for, 194
viewport transformations, 97, 102, 125–128
  photograph analogy, 94
  rendering pipeline stage, 12, 532
visual simulation
  fog, use of, 239
Voronoi polygons, 524

W

w coordinates, 37, 96, 102
  avoiding negative values, 601
  lighting, use with, 183
  perspective division, 128, 532
warping images, 523
Watt, Alan, 318
web sites
  IBM OS/2 software and documentation, 570
  Microsoft Developer Network, 574
  Silicon Graphics' OpenGL, xxii
  Template Graphics Software, 566
WGL, 14, 574
  wglCopyContext(), 575, 576
  wglCreateContext(), 574, 575, 576
  wglCreateLayerContext(), 575, 576
  wglDeleteContext(), 576
  wglDescribeLayerPlane(), 574, 576
  wglDestroyContext(), 575
  wglGetCurrentContext(), 575, 577
  wglGetCurrentDC(), 575, 577
  wglGetLayerPaletteEntries(), 576, 577
  wglMakeCurrent(), 575, 577
  wglRealizeLayerPalette(), 575, 577
  wglShareLists(), 575, 576
  wglSwapLayerBuffers(), 575, 577
  wglUseFontBitmaps(), 576, 577
  wglUseFontOutlines(), 576, 577

Williams, Lance, 338
Win32
  ChoosePixelFormat(), 574, 576
  CreateDIBitmap(), 575, 577
  CreateDIBSection(), 575, 577
  DeleteObject(), 575, 577
  DescribePixelFormat(), 574, 576
  GetVersion(), 574, 576
  GetVersionEx(), 574, 576
  SetPixelFormat(), 574, 576
  SwapBuffers(), 575, 577
winding rules, 418–422
  computational solid geometry, used for, 421
  reversing winding direction, 422
window coordinates, 97, 126
  feedback mode, 493
  polygon offset, 248
window management
  glViewport() called, when window resized, 126
  using GLUT, 16, 35
working set of textures, 332, 346, 351
  fragmentation of texture memory, 353
writemask, See masking (buffers)
writing pixel data, See pixel data (drawing)

X

X Window System, 14, 562
  client-server rendering, 5
  minimum framebuffer configuration, 376
  X Visual, 162, 562

Z

z buffer, See depth buffer
z coordinates, See depth coordinates
zooming images, 305
  filtered, 528