Contents

List of Figures xiii

1. **Introduction** 1
   - 1.1 Application fields 1
   - 1.2 From a real scene to an image 3
   - 1.3 Organisation of the book 4

2. **Basic principles of two-dimensional graphics** 7
   - 2.1 Raster versus vector graphics 7
   - 2.2 The first Java 2D program 10
   - 2.3 Basic geometric objects 14
   - 2.4 Basic geometric objects in Java 2D 17
   - 2.5 Geometric transformations 23
   - 2.6 Homogeneous coordinates 28
   - 2.7 Applications of transformations 31
   - 2.8 Geometric transformations in Java 2D 33
   - 2.9 Animation and movements based on transformations 37
   - 2.10 Movements via transformations in Java 2D 39
   - 2.11 Interpolators for continuous changes 41
   - 2.12 Implementation of interpolators in Java 2D 44
   - 2.13 Single or double precision 45
   - 2.14 Exercises 48

3. **Drawing lines and curves** 49
   - 3.1 Lines and pixel graphics 49
   - 3.2 The midpoint algorithm for lines 52

3.3 Structural algorithms 60
3.4 Pixel densities and line styles 63
3.4.1 Different line styles with Java 2D 66
3.5 Line clipping 67
3.6 The midpoint algorithm for circles 75
3.7 Drawing arbitrary curves 79
3.8 Antialiasing 80
3.8.1 Antialiasing with Java 2D 82
3.9 Drawing thick lines 83
3.9.1 Drawing thick lines with Java 2D 84
3.10 Exercises 86

4. Areas, text and colours 87
4.1 Filling areas 87
4.2 Buffered images in Java 2D 91
4.2.1 Double buffering in Java 2D 92
4.2.2 Loading and saving of images with Java 2D 94
4.2.3 Textures in Java 2D 95
4.3 Displaying text 96
4.4 Text in Java 2D 97
4.5 Grey images and intensities 99
4.6 Colour models 101
4.6.1 Colours in Java 2D 106
4.7 Colour interpolation 107
4.8 Colour interpolation with Java 2D 110
4.9 Exercises 112

5. Basic principles of three-dimensional graphics 113
5.1 From a 3D world to a model 113
5.2 Geometric transformations 115
5.2.1 Java 3D 118
5.2.2 Geometric transformations in Java 3D 119
5.3 The scenegraph 120
5.4 Elementary geometric objects in Java 3D 123
5.5 The scenegraph in Java 3D 124
5.6 Animation and moving objects 130
5.7 Animation in Java 3D 133
5.8 Projections 139
5.8.1 Projections in Java 3D 146
5.9 Exercises 147

6. Modelling three-dimensional objects 149
6.1 Three-dimensional objects and their surfaces 149
6.2 Topological notions 152
6.3 Modelling techniques 154
6.4 Surface modelling with polygons in Java 3D 159
6.5 Importing geometric objects into Java 3D 162
6.6 Parametric curves and freeform surfaces 163
6.6.1 Parametric curves 164
6.6.2 Efficient computation of polynomials 170
6.6.3 Freeform surfaces 171
6.7 Normal vectors for surfaces 173
6.7.1 Normal vectors in Java 3D 176
6.8 Exercises 178

7. Visible surface determination 179
7.1 The clipping volume 179
7.1.1 Clipping in Java 3D 182
7.2 Principles of algorithms for visible surface determination 183
7.2.1 Image-precision and object-precision algorithms 183
7.2.2 Back-face culling 184
7.2.3 Spatial partitioning 186
7.3 Image-precision techniques 187
7.3.1 The z-buffer algorithm 187
7.3.2 Scan line technique for edges 190
7.3.3 Ray casting 192
7.4 Priority algorithms 195
7.5 Exercises 199

8. Illumination and shading 201
8.1 Light sources 202
8.2 Light sources in Java 3D 206
8.3 Reflection 208
8.4 Shading in Java 3D 216
8.5 Shading 218
8.5.1 Constant and Gouraud shading in Java 3D 222
8.6 Shadows 222
8.7 Transparency 224
8.7.1 Transparency in Java 3D 226
8.8 Textures 227
8.9 Textures in Java 3D 229
8.10 The radiosity model 231
8.11 Ray tracing 236

8.12 Exercises 238

9. Special effects and virtual reality 239

9.1 Fog and particle systems 240

9.2 Fog in Java 3D 242

9.3 Dynamic surfaces 243

9.4 Interaction 245

9.5 Interaction in Java 3D 245

9.6 Collision detection 249

9.7 Collision detection in Java 3D 250

9.8 Sound effects 256

9.9 Sound effects in Java 3D 257

9.10 Stereoscopic viewing 258

9.11 Exercises 263

Appendix: Useful links 265

Appendix: Example programs 267

Appendix: References to Java 2D classes and methods 273

Appendix: References to Java 3D classes and methods 275

Bibliography 277

Index 281

List of Figures

1.1 From a scene to an image 3

2.1 Original image, vector and pixel graphics 8
2.2 The tip of an arrow drawn as raster graphics in two different resolutions 9
2.3 An alternative representation for pixels 10
2.4 The Java 2D API extends AWT 11
2.5 The result of the first Java 2D program 11
2.6 A self-overlapping, a nonconvex and a convex polygon 14
2.7 Definition of quadratic and cubic curves 15
2.8 Fitting a cubic curve to a line without sharp bends 16
2.9 Union, intersection, difference and symmetric difference of a circle and a rectangle 17
2.10 An example for a GeneralPath 19
2.11 An example for a rectangle and an ellipse 20
2.12 An arc of an ellipse, a segment and an arc with its corresponding chord 22
2.13 Scaling applied to a rectangle 24
2.14 A rotation applied to a rectangle 26
2.15 A shear transformation applied to a rectangle 26
2.16 Translation of a rectangle 27
2.17 Homogeneous coordinates 29
2.18 Differing results on changing the order for the application of a translation and a rotation 30
2.19 From world to window coordinates 32
2.20 A moving clock with a rotating hand 38

2.21 Changing one ellipse to another by convex combinations of transformations . . . 42
2.22 Two letters each defined by five points and two quadratic curves . . . 43
2.23 Stepwise transformation of two letters into each other . . . 44

3.1 Pseudocode for a naïve line drawing algorithm . . . 50
3.2 Lines resulting from the naïve line drawing algorithm . . . 50
3.3 The two candidates for the next pixel for the line drawing algorithm 53
3.4 The new midpoint depending on whether the previously drawn pixel was E or NE . . . 57
3.5 Drawing a line with the Bresenham algorithm . . . 60
3.6 A repeated pixel pattern for drawing a line on pixel raster . . . 61
3.7 Different pixel densities depending on the slope of a line . . . 64
3.8 Different line styles . . . 65
3.9 Different dash lengths for the same bitmask . . . 66
3.10 Examples for different line styles . . . 67
3.11 Different cases for line clipping . . . 68
3.12 Bit code for Cohen-Sutherland clipping . . . 70
3.13 Cohen-Sutherland line clipping . . . 72
3.14 Cyrus-Beck line clipping . . . 73
3.15 Potential intersection points with the clipping rectangle . . . 74
3.16 Finding the pixel where a line enters the clipping rectangle . . . 75
3.17 Exploiting symmetry for drawing circles . . . 76
3.18 Midpoint algorithm for circles . . . 76
3.19 Drawing arbitrary curves . . . 80
3.20 Unweighted area sampling . . . 81
3.21 Estimation of the area by sampling with refined pixels . . . 81
3.22 Weighted area sampling . . . 81
3.23 Pixel replication and the moving pen technique . . . 83
3.24 Different line endings and joins . . . 84

4.1 Odd parity rule . . . 88
4.2 Scan line technique for filling polygons . . . 89
4.3 A scan line intersecting two vertices of a polygon . . . 89
4.4 Filling a polygon can lead to aliasing effects . . . 90
4.5 Filling an area with a texture . . . 90
4.6 Italic and boldface printing for letters given in raster graphics . . . 97
4.7 Grey-level representation based on halftoning for a 2×2 (top line) and on 3×3 pixel matrices (3 bottom lines) . . . 100
4.8 Distribution of the energies over the wavelengths for high (left) and low (right) saturation . . . 102
4.9 RGB and CMY model . . . 103

4.10 HSV model 105
4.11 HLS model 105
4.12 Compatible triangulations of two images 108
4.13 Computation of the interpolated colour of a pixel 110

5.1 A right-handed coordinate system 115
5.2 A chair constructed with elementary geometric objects 120
5.3 A scene composed of various elementary objects 121
5.4 The scenegraph for figure 5.3 122
5.5 The overall scenegraph for Java 3D 125
5.6 Excerpt of the scenegraph with dynamic transformations 131
5.7 Progression of the Alpha-values 134
5.8 Perspective and parallel projection 140
5.9 Mapping an arbitrary plane to a plane parallel to the x/y-plane 141
5.10 Derivation of the matrix for the perspective projection 141
5.11 Vanishing point for perspective projection 145
5.12 One-, two- and three-point perspective projections 145

6.1 Isolated and dangling edges and faces 150
6.2 Triangulation of a polygon 150
6.3 Orientation of polygons 151
6.4 A tetrahedron 152
6.5 A set $M \subset \mathbb{R}^2$ of points, its interior, boundary, closure and regularisation 152
6.6 Modelling a three-dimensional object with voxels 154
6.7 Recursive partition of an area into squares 155
6.8 The quadtree for figure 6.7 156
6.9 An object that was constructed using elementary geometric objects and set-theoretic operations shown on the right 157
6.10 Two objects and their sweep representations 157
6.11 Tesselation of the helicopter scene in figure 5.3 158
6.12 Representation of a sphere with different tesselations 158
6.13 Two curves obtained from a surface that is scanned along the coordinate axes 164
6.14 An interpolation polynomial of degree 5 defined by the control points $(0,0), (1,0), (2,0), (3,0), (4,1), (5,0)$ 165
6.15 B-spline with knots P_1, P_4, P_7 and inner Bézier points P_2, P_3, P_5, P_6 168
6.16 Condition for the inner Bézier points for a twice differentiable, cubic B-spline 169
6.17 A parametric freeform surface 172
6.18 A net of Bézier points for the definition of a Bézier surface 173
6.19 A triangular grid for the definition of a Bézier surface 174

6.20 Normal vectors to the original surface in the vertices of an approximating triangle 175
6.21 Interpolated and noninterpolated normal vectors 177

7.1 The angle α determines the range on the projection plane that corresponds to the width of the display window 180
7.2 The clipping volume for parallel projection (top) and perspective projection (bottom) 181
7.3 A front face whose normal vector forms an acute angle with the direction of projection and a back face whose normal vector forms an obtuse angle with the direction of projection 185
7.4 Partitioning of the clipping volume for image-precision (left) and object-precision algorithms (right) 186
7.5 Principle of the z-buffer algorithm 188
7.6 Determining the active edges for the scan lines v_1, v_2, v_3, v_4 191
7.7 Ray casting 192
7.8 Projection of a polygon to decide whether a point lies within the polygon 193
7.9 Supersampling 194
7.10 No overlap in the x-coordinate (left) or the y-coordinate (right) 196
7.11 Does one polygon lie completely in front or behind the plane induced by the other? 196
7.12 Determining whether a polygon lies completely in front of the plane induced by the other polygon 197
7.13 A case where no correct order exists in which the polygons should be projected 197

8.1 Objects with and without illumination and shading effects 201
8.2 Cone of light from a spotlight 204
8.3 The Warn model for a spotlight 205
8.4 The functions $(\cos\gamma)^{64}$, $(\cos\gamma)^8$, $(\cos\gamma)^2$, $\cos\gamma$ 205
8.5 Light intensity depending on the angle of the light 210
8.6 Diffuse reflection 211
8.7 Diffuse and specular reflection 212
8.8 Computation of ideal specular reflection 213
8.9 The halfway vector **h** in Phong's model 215
8.10 A sphere in different tesselations rendered with flat shading 218
8.11 The colour intensity as a function over a triangle for Gouraud shading 219
8.12 Scan line technique for the computation of Gouraud shading 220
8.13 Interpolated normal vectors for Phong shading 221
8.14 Shadow on an object 223
8.15 50% (left) and 25% (right) screen-door transparency 225

8.16 Using a texture 227
8.17 Modelling a mirror by a reflection mapping 228
8.18 Bump mapping 229
8.19 Illumination among objects 232
8.20 Determination of the form factors 234
8.21 Determination of the form factors according to Nusselt 235
8.22 Recursive ray tracing 237

9.1 Linear and exponential fog 241
9.2 Skeleton and skinning 244
9.3 Bounding volume in the form of a cube and a sphere 250
9.4 Parallax and accommodation for natural and artificial stereoscopic viewing 261
9.5 Parallax for stereoscopic viewing 261