

1. Record Nr.	UNINA9910153249703321
Autore	Angel Edward
Titolo	Interactive computer graphics : a top-down approach with WebGL // Edward Angel, Dave Schreiner
Pubbl/distr/stampa	Boston : , : Pearson, , 2015
ISBN	1-292-01933-6
Edizione	[Seventh edition, Global edition.]
Descrizione fisica	1 online resource (732 pages) : illustrations
Disciplina	006.6
Soggetti	Computer graphics - Computer programs
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di bibliografia	Includes bibliographical references and indexes.
Nota di contenuto	Cover -- Dedication -- Contents -- Preface -- Chapter 1: Graphics Systems and Models -- 1.1 Applications of Computer Graphics -- 1.1.1 Display of Information -- 1.1.2 Design -- 1.1.3 Simulation and Animation -- 1.1.4 User Interfaces -- 1.2 A Graphics System -- 1.2.1 Pixels and the Framebuffer -- 1.2.2 The CPU and the GPU -- 1.2.3 Output Devices -- 1.2.4 Input Devices -- 1.3 Images: Physical and Synthetic -- 1.3.1 Objects and Viewers -- 1.3.2 Light and Images -- 1.3.3 Imaging Models -- 1.4 Imaging Systems -- 1.4.1 The Pinhole Camera -- 1.4.2 The Human Visual System -- 1.5 The Synthetic-Camera Model -- 1.6 The Programmer's Interface -- 1.6.1 The Pen-Plotter Model -- 1.6.2 Three-Dimensional APIs -- 1.6.3 A Sequence of Images -- 1.6.4 The Modeling-Rendering Paradigm -- 1.7 Graphics Architectures -- 1.7.1 Display Processors -- 1.7.2 Pipeline Architectures -- 1.7.3 The Graphics Pipeline -- 1.7.4 Vertex Processing -- 1.7.5 Clipping and Primitive Assembly -- 1.7.6 Rasterization -- 1.7.7 Fragment Processing -- 1.8 Programmable Pipelines -- 1.9 Performance Characteristics -- 1.10 OpenGL Versions and WebGL -- Summary and Notes -- Suggested Readings -- Exercises -- Chapter 2: Graphics Programming -- 2.1 The Sierpinski Gasket -- 2.2 Programming Two-Dimensional Applications -- 2.3 The WebGL Application Programming Interface -- 2.3.1 Graphics Functions -- 2.3.2 The Graphics Pipeline and State Machines -- 2.3.3 OpenGL and WebGL -- 2.3.4 The WebGL Interface -- 2.3.5 Coordinate Systems -- 2.4 Primitives and Attributes -- 2.4.1 Polygon Basics -- 2.4.2 Polygons

in WebGL -- 2.4.3 Approximating a Sphere -- 2.4.4 Triangulation -- 2.4.5 Text -- 2.4.6 Curved Objects -- 2.4.7 Attributes -- 2.5 Color -- 2.5.1 RGB Color -- 2.5.2 Indexed Color -- 2.5.3 Setting of Color Attributes -- 2.6 Viewing -- 2.6.1 The Orthographic View -- 2.6.2 Two-Dimensional Viewing.

2.7 Control Functions -- 2.7.1 Interaction with the Window System -- 2.7.2 Aspect Ratio and Viewports -- 2.7.3 Application Organization -- 2.8 The Gasket Program -- 2.8.1 Sending Data to the GPU -- 2.8.2 Rendering the Points -- 2.8.3 The Vertex Shader -- 2.8.4 The Fragment Shader -- 2.8.5 Combining the Parts -- 2.8.6 The `initShaders` Function -- 2.8.7 The `init` Function -- 2.8.8 Reading the Shaders from the Application -- 2.9 Polygons and Recursion -- 2.10 The Three-Dimensional Gasket -- 2.10.1 Use of Three-Dimensional Points -- 2.10.2 Naming Conventions -- 2.10.3 Use of Polygons in Three Dimensions -- 2.10.4 Hidden-Surface Removal -- Summary and Notes -- Suggested Readings -- Exercises -- Chapter 3: Interaction and Animation -- 3.1 Animation -- 3.1.1 The Rotating Square -- 3.1.2 The Display Process -- 3.1.3 Double Buffering -- 3.1.4 Using a Timer -- 3.1.5 Using `requestAnimationFrame` -- 3.2 Interaction -- 3.3 Input Devices -- 3.4 Physical Input Devices -- 3.4.1 Keyboard Codes -- 3.4.2 The Light Pen -- 3.4.3 The Mouse and the Trackball -- 3.4.4 Data Tablets, Touch Pads, and Touch Screens -- 3.4.5 The Joystick -- 3.4.6 Multidimensional Input Devices -- 3.4.7 Logical Devices -- 3.4.8 Input Modes -- 3.5 Clients and Servers -- 3.6 Programming Event-Driven Input -- 3.6.1 Events and Event Listeners -- 3.6.2 Adding a Button -- 3.6.3 Menus -- 3.6.4 Using Keycodes -- 3.6.5 Sliders -- 3.7 Position Input -- 3.8 Window Events -- 3.9 Picking -- 3.10 Building Models Interactively -- 3.11 Design of Interactive Programs -- Summary and Notes -- Suggested Readings -- Exercises -- Chapter 4: Geometric Objects and Transformations -- 4.1 Scalars, Points, and Vectors -- 4.1.1 Geometric Objects -- 4.1.2 Coordinate-Free Geometry -- 4.1.3 The Mathematical View: Vector and Affine Spaces -- 4.1.4 The Computer Science View -- 4.1.5 Geometric ADTs -- 4.1.6 Lines -- 4.1.7 Affine Sums.

4.1.8 Convexity -- 4.1.9 Dot and Cross Products -- 4.1.10 Planes -- 4.2 Three-Dimensional Primitives -- 4.3 Coordinate Systems and Frames -- 4.3.1 Representations and N-Tuples -- 4.3.2 Change of Coordinate Systems -- 4.3.3 Example: Change of Representation -- 4.3.4 Homogeneous Coordinates -- 4.3.5 Example: Change in Frames -- 4.3.6 Working with Representations -- 4.4 Frames in WebGL -- 4.5 Matrix and Vector Types -- 4.5.1 Row versus Column Major Matrix Representations -- 4.6 Modeling a Colored Cube -- 4.6.1 Modeling the Faces -- 4.6.2 Inward- and Outward-Pointing Faces -- 4.6.3 Data Structures for Object Representation -- 4.6.4 The Colored Cube -- 4.6.5 Color Interpolation -- 4.6.6 Displaying the Cube -- 4.6.7 Drawing with Elements -- 4.7 Affine Transformations -- 4.8 Translation, Rotation, and Scaling -- 4.8.1 Translation -- 4.8.2 Rotation -- 4.8.3 Scaling -- 4.9 Transformations in Homogeneous Coordinates -- 4.9.1 Translation -- 4.9.2 Scaling -- 4.9.3 Rotation -- 4.9.4 Shear -- 4.10 Concatenation of Transformations -- 4.10.1 Rotation About a Fixed Point -- 4.10.2 General Rotation -- 4.10.3 The Instance Transformation -- 4.10.4 Rotation About an Arbitrary Axis -- 4.11 Transformation Matrices in WebGL -- 4.11.1 Current Transformation Matrices -- 4.11.2 Basic Matrix Functions -- 4.11.3 Rotation, Translation, and Scaling -- 4.11.4 Rotation About a Fixed Point -- 4.11.5 Order of Transformations -- 4.12 Spinning of the Cube -- 4.12.1 Uniform Matrices -- 4.13 Interfaces to Three-Dimensional Applications -- 4.13.1 Using Areas of the Screen -- 4.13.2 A Virtual

Trackball -- 4.13.3 Smooth Rotations -- 4.13.4 Incremental Rotation -- 4.14 Quaternions -- 4.14.1 Complex Numbers and Quaternions -- 4.14.2 Quaternions and Rotation -- 4.14.3 Quaternions and Gimbal Lock -- Summary and Notes -- Suggested Readings -- Exercises -- Chapter 5: Viewing.

5.1 Classical and Computer Viewing -- 5.1.1 Classical Viewing -- 5.1.2 Orthographic Projections -- 5.1.3 Axonometric Projections -- 5.1.4 Oblique Projections -- 5.1.5 Perspective Viewing -- 5.2 Viewing with a Computer -- 5.3 Positioning of the Camera -- 5.3.1 Positioning of the Camera Frame -- 5.3.2 Two Viewing APIs -- 5.3.3 The Look-At Function -- 5.3.4 Other Viewing APIs -- 5.4 Parallel Projections -- 5.4.1 Orthogonal Projections -- 5.4.2 Parallel Viewing with WebGL -- 5.4.3 Projection Normalization -- 5.4.4 Orthogonal Projection Matrices -- 5.4.5 Oblique Projections -- 5.4.6 An Interactive Viewer -- 5.5 Perspective Projections -- 5.5.1 Simple Perspective Projections -- 5.6 Perspective Projections with WebGL -- 5.6.1 Perspective Functions -- 5.7 Perspective Projection Matrices -- 5.7.1 Perspective Normalization -- 5.7.2 WebGL Perspective Transformations -- 5.7.3 Perspective Example -- 5.8 Hidden-Surface Removal -- 5.8.1 Culling -- 5.9 Displaying Meshes -- 5.9.1 Displaying Meshes as Surfaces -- 5.9.2 Polygon Offset -- 5.9.3 Walking through a Scene -- 5.10 Projections and Shadows -- 5.10.1 Projected Shadows -- 5.11 Shadow Maps -- Summary and Notes -- Suggested Readings -- Exercises -- Chapter 6: Lighting and Shading -- 6.1 Light and Matter -- 6.2 Light Sources -- 6.2.1 Color Sources -- 6.2.2 Ambient Light -- 6.2.3 Point Sources -- 6.2.4 Spotlights -- 6.2.5 Distant Light Sources -- 6.3 The Phong Reflection Model -- 6.3.1 Ambient Reflection -- 6.3.2 Diffuse Reflection -- 6.3.3 Specular Reflection -- 6.3.4 The Modified Phong Model -- 6.4 Computation of Vectors -- 6.4.1 Normal Vectors -- 6.4.2 Angle of Reflection -- 6.5 Polygonal Shading -- 6.5.1 Flat Shading -- 6.5.2 Smooth and Gouraud Shading -- 6.5.3 Phong Shading -- 6.6 Approximation of a Sphere by Recursive Subdivision -- 6.7 Specifying Lighting Parameters -- 6.7.1 Light Sources -- 6.7.2 Materials. 6.8 Implementing a Lighting Model -- 6.8.1 Applying the Lighting Model in the Application -- 6.8.2 Efficiency -- 6.8.3 Lighting in the Vertex Shader -- 6.9 Shading of the Sphere Model -- 6.10 Per-Fragment Lighting -- 6.11 Nonphotorealistic Shading -- 6.12 Global Illumination -- Summary and Notes -- Suggested Readings -- Exercises -- Chapter 7: Discrete Techniques -- 7.1 Buffers -- 7.2 Digital Images -- 7.3 Mapping Methods -- 7.4 Two-Dimensional Texture Mapping -- 7.5 Texture Mapping in WebGL -- 7.5.1 Texture Objects -- 7.5.2 The Texture Image Array -- 7.5.3 Texture Coordinates and Samplers -- 7.5.4 Texture Sampling -- 7.5.5 Working with Texture Coordinates -- 7.5.6 Multitexturing -- 7.6 Texture Generation -- 7.7 Environment Maps -- 7.8 Reflection Map Example -- 7.9 Bump Mapping -- 7.9.1 Finding Bump Maps -- 7.9.2 Bump Map Example -- 7.10 Blending Techniques -- 7.10.1 Opacity and Blending -- 7.10.2 Image Blending -- 7.10.3 Blending in WebGL -- 7.10.4 Antialiasing Revisited -- 7.10.5 Back-to-Front and Front-to-Back Rendering -- 7.10.6 Scene Antialiasing and Multisampling -- 7.10.7 Image Processing -- 7.10.8 Other Multipass Methods -- 7.11 GPGPU -- 7.12 Framebuffer Objects -- 7.13 Buffer Ping-Ponging -- 7.14 Picking -- Summary and Notes -- Suggested Readings -- Exercises -- Chapter 8: From Geometry To Pixels -- 8.1 Basic Implementation Strategies -- 8.2 Four Major Tasks -- 8.2.1 Modeling -- 8.2.2 Geometry Processing -- 8.2.3 Rasterization -- 8.2.4 Fragment Processing -- 8.3 Clipping -- 8.4 Line-Segment Clipping -- 8.4.1 Cohen-Sutherland Clipping -- 8.4.2 Liang-Barsky Clipping -- 8.5 Polygon Clipping -- 8.6 Clipping of

Other Primitives -- 8.6.1 Bounding Boxes and Volumes -- 8.6.2
Curves, Surfaces, and Text -- 8.6.3 Clipping in the Framebuffer -- 8.7
Clipping in Three Dimensions -- 8.8 Rasterization -- 8.9 Bresenham's
Algorithm.
8.10 Polygon Rasterization.

Sommario/riassunto

This book is suitable for undergraduate students in computer science and engineering, for students in other disciplines who have good programming skills, and for professionals. Computer animation and graphics are now prevalent in everyday life from the computer screen, to the movie screen, to the smart phone screen. The growing excitement about WebGL applications and their ability to integrate HTML5, inspired the authors to exclusively use WebGL in the Seventh Edition of Interactive Computer Graphics with WebGL. This is the only introduction to computer graphics text for undergraduates that fully integrates WebGL and emphasizes application-based programming. The top-down, programming-oriented approach allows for coverage of engaging 3D material early in the course so students immediately begin to create their own 3D graphics. Teaching and Learning Experience This program will provide a better teaching and learning experience-for you and your students. It will help: Engage Students Immediately with 3D Material: A top-down, programming-oriented approach allows for coverage of engaging 3D material early in the course so students immediately begin to create their own graphics. Introduce Computer Graphics Programming with WebGL and JavaScript: WebGL is not only fully shader-based-each application must provide at least a vertex shader and a fragment shader-but also a version that works within the latest web browsers.
