Computer Graphics: Programming, Problem Solving, and Visual Communication

Dr. Steve Cunningham
Computer Science Department
California State University Stanislaus
Turlock, CA  95382

copyright © 2002, Steve Cunningham
All rights reserved
These notes are a draft of a textbook for an introductory computer graphics course that emphasizes graphics programming and is intended for undergraduate students who have a sound background in programming. Its goal is to introduce fundamental concepts and processes for computer graphics, give students experience in computer graphics programming using the OpenGL application programming interface (API), and show the power of visual communication and of computer graphics in the sciences.

The contents below represent a relatively mature version of these notes, although some reorganization of the material is still expected and some additional topics may be developed. We hope you will find these notes to be helpful in learning computer graphics and encourage you to give us feedback with the feedback pages at the end of the notes.

**CONTENTS:**

**Getting Started**
- What is a graphics API?
- Overview of the book
- What is computer graphics?
- The 3D Graphics Pipeline
  - 3D model coordinate systems
  - 3D world coordinate system
  - 3D eye coordinate system
  - Clipping
  - Projections
  - 2D eye coordinates
  - 2D screen coordinates
  - Overall viewing process
  - Different implementation, same result
  - Summary of viewing advantages
- A basic OpenGL program
  - The structure of the main() program using OpenGL
    - Model space
    - Modeling transformation
    - 3D world space
    - Viewing transformation
    - 3D eye space
    - Projections
    - 2D eye space
    - 2D screen space
    - Another way to see the program
- OpenGL extensions

**Chapter 1: Viewing and Projection**
- Introduction
- Fundamental model of viewing
- Definitions
  - Setting up the viewing environment
  - Defining the projection
  - View volumes
  - Calculating the perspective transformation
  - Defining the window and viewport
- Some aspects of managing the view
  - Hidden surfaces
- Double buffering
- Clipping planes
- Stereo viewing
- Implementation of viewing and projection in OpenGL
  - Defining a window and viewport
  - Reshaping the window
  - Defining a viewing environment
  - Defining perspective projection
  - Defining an orthogonal projection
  - Managing hidden surface viewing
  - Setting double buffering
  - Defining clipping planes
- Implementing a stereo view
- Questions
- Exercises
- Experiments

Chapter 2: Principles of Modeling
- Introduction
  Simple Geometric Modeling
  - Introduction
  - Definitions
  - Some examples
    - Point and points
    - Line segments
    - Connected lines
    - Triangle
    - Sequence of triangles
    - Quadrilateral
    - Sequence of quads
    - General polygon
    - Polyhedron
    - Aliasing and antialiasing
    - Normals
    - Data structures to hold objects
    - Additional sources of graphic objects
    - A word to the wise
Transformations and modeling
- Introduction
  - Definitions
    - Transformations
    - Composite transformations
    - Transformation stacks and their manipulation
    - Compiling geometry
  - A word to the wise
Scene graphs and modeling graphs
- Introduction
  - A brief summary of scene graphs
    - An example of modeling with a scene graph
  - The viewing transformation
  - The scene graph and depth testing
  - Using the modeling graph for coding
    - Example
    - Using standard objects to create more complex scenes
Chapter 3: Implementing Modeling in OpenGL

- The OpenGL model for specifying geometry
  - Point and points mode
  - Line segments
  - Line strips
  - Line loops
  - Triangle
  - Sequence of triangles
  - Quads
  - Quad strips
  - General polygon
  - Antialiasing
  - The cube we will use in many examples
- Additional objects with the OpenGL toolkits
  - GLU quadric objects
    - GLU cylinder
    - GLU disk
    - GLU sphere
  - The GLUT objects
  - An example
- A word to the wise
- Transformations in OpenGL
- Code examples for transformations
  - Simple transformations
  - Transformation stacks
  - Inverting the eyepoint transformation
  - Creating display lists

Chapter 4: Mathematics for Modeling

- Coordinate systems and points
- Points, lines, and line segments
- Distance from a point to a line
- Line segments and parametric curves
- Vectors
- Dot and cross products of vectors
- Reflection vectors
- Transformations
- Planes and half-spaces
- Distance from a point to a plane
- Polygons and convexity
- Polyhedra
- Collision detection
- Polar, cylindrical, and spherical coordinates
- Higher dimensions?

Chapter 5: Color and Blending

- Introduction
- Definitions
  - The RGB cube
  - Luminance
- Other color models
- Color depth
- Color gamut
- Color blending with the alpha channel
- Challenges in blending
- Modeling transparency with blending
- Indexed color
- Using color to create 3D images

• Some examples
  - An object with partially transparent faces
• Color in OpenGL
  - Enabling blending
• A word to the wise
• Code examples
  - A model with parts having a full spectrum of colors
  - The HSV cone
  - The HLS double cone
  - An object with partially transparent faces
• Questions
• Exercises
• Experiments

Chapter 6: Visual Communication
• Introduction
• General issues in visual communication
  - Use appropriate representation for your information
  - Keep your images focused
  - Use appropriate presentation levels for your information
  - Use appropriate forms for your information
  - Be very careful to be accurate with your information
  - Understand and respect the cultural context of your audience
  - Make your interactions reflect familiar and comfortable relationships between cause and effect
• Shape
  - Comparing shape and color encodings
• Color
  - Emphasis colors
  - Background colors
  - Color deficiencies in audience
  - Naturalistic color
  - Pseudocolor and color ramps
  - Implementing color ramps
  - Using color ramps
  - To light or not to light
  - Higher dimensions
• Dimensions
• Image context
  - Choosing an appropriate view
  - Legends to help communicate your encodings
  - Labels to help communicate your problem
• Motion
  - Leaving traces of motion
  - Motion blurring
• Interactions
• Cultural context of the audience
• Accuracy
• Output media
• Implementing some of these ideas in OpenGL
  - Using color ramps
  - Legends and labels
  - Creating traces
  - Using the accumulation buffer
• A word to the wise

Chapter 7: Graphical Problem Solving in Science
• Introduction
• Examples
• Diffusion
  - Temperatures in a bar
  - Spread of disease
• Function graphing and applications
• Parametric curves and surfaces
• Graphical objects that are the results of limit processes
• Scalar fields
• Representation of objects and behaviors
  - Gas laws and diffusion principles
  - Molecular display
  - Monte Carlo modeling process
• 4D graphing
  - Volume data
  - Vector fields
• Graphing in higher dimensions
• Data-driven graphics
• Code examples
  - Diffusion
  - Function graphing
  - Parametric curves and surfaces
  - Limit processes
  - Scalar fields
  - Representation of objects and behaviors
  - Molecular display
  - Monte Carlo modeling
  - 4D graphing
  - Higher dimensional graphing
  - Data-driven graphics
• Credits

Chapter 8: The Rendering Pipeline
• Introduction
• The pipeline
• The rendering pipeline for OpenGL
  - Texture mapping in the rendering pipeline
  - Per-fragment operations
  - Some extensions to OpenGL
  - An implementation of the rendering pipeline in a graphics card
• The rasterization process
Chapter 9: Lighting and Shading

Lighting
• Definitions
  - Ambient, diffuse, and specular light
  - Surface normals
• Light properties
  - Positional lights
  - Spotlights
  - Attenuation
  - Directional lights
  - Positional and moving lights

Materials

Shading
• Definitions
  - Flat shading
  - Smooth shading
• Examples of flat and smooth shading
• Calculating per-vertex normals
  - Averaging polygon normals
  - Analytic computations
• Other shading models
  - Vertex and pixel shaders

Global Illumination

Local Illumination
• Lights and materials in OpenGL
  - Specifying and defining lights
  - Defining materials
  - Setting up a scene to use lighting
  - Using GLU quadric objects
  - An example: lights of all three primary colors applied to a white surface
  - Code for the example
• A word to the wise
  - Shading example

Questions

Exercises

Experiments

Chapter 10: Event Handling

• Introduction
• Definitions
• Some examples of events
  - keypress events
  - mouse events
  - menu events
  - window events
  - system events
  - software events
• The vocabulary of interaction
• A word to the wise
• Events in OpenGL
• Callback registering
• Some details
• Code examples
  - Idle event callback
- Keyboard callback
- Menu callback
- Mouse callback for object selection
- Mouse callback for mouse motion

**The MUI (Micro User Interface) Facility**

- Introduction
- Definitions
  - Menu bars
  - Buttons
  - Radio buttons
  - Text boxes
  - Horizontal sliders
  - Vertical sliders
  - Text labels
- Using the MUI functionality
- Some examples
- Installing MUI for Windows
- A word to the wise

**Chapter 11: Texture Mapping**

- Introduction
- Definitions
  - 1D texture maps
  - 2D texture maps
  - 3D texture maps
  - Associating a vertex with a texture point
  - The relation between the color of the object and the color of the texture map
  - Other meanings for texture maps
- Creating a texture map
  - Getting an image as a texture map
  - Generating a synthetic texture map
- Texture mapping and billboards
- Interpolation for texture maps
- Antialiasing in texturing
- MIP mapping
- Multitexturing
- Using billboards
- Texture mapping in OpenGL
  - Associating vertices and texture points
  - Capturing a texture from the screen
  - Texture environment
  - Texture parameters
  - Getting and defining a texture map
  - Texture coordinate control
  - Texture interpolation
  - Texture mapping and GLU quadrics
- Some examples
  - The Chromadepth™ process
  - Using 2D texture maps to add interest to a surface
  - Environment maps
- A word to the wise
- Code examples
  - A 1D color ramp
  - An image on a surface
Chapter 12: Dynamics and Animation
- Introduction
- Definitions
- Keyframe animation
  - Temporal aliasing
  - Building an animation
- Some examples
  - Moving objects in your model
  - Moving parts of objects in your model
  - Moving the eye point or the view frame in your model
  - Changing features of your models
- Some points to consider when doing animations with OpenGL
- Code examples
- A word to the wise

Chapter 13: High-Performance Graphics Techniques
- Definitions
- Techniques
  - Hardware avoidance
  - Designing out visible polygons
  - Culling polygons
  - Avoiding depth comparisons
  - Front-to-back drawing
  - Binary space partitioning
  - Clever use of textures
  - System speedups
  - Level of detail
  - Reducing lighting computation
  - Fog
  - Collision detection
- A word to the wise

Chapter 14: Object Selection
- Introduction
- Picking in OpenGL
- Definitions
- Making picking work
- The pick matrix
- Using the back color buffer to do picking
- A selection example
- A word to the wise
- Questions
- Exercises
- Experiments

Chapter 15: Interpolation and Spline Modeling
- Introduction
  - Interpolations
- Extending interpolations to more control points
  - Interpolations in OpenGL
  - Automatic normal and texture generation with evaluators
  - Additional techniques
  - Definitions
  - Some examples
    - Spline curves
    - Spline surfaces
  - A word to the wise

Chapter 16: Per-Pixel Operations
  - Introduction
  - Definitions
  - Ray casting
  - Ray tracing
  - Ray tracing and global illumination
  - Volume rendering
  - Fractal images
  - Iterated function systems
  - Per-pixel operations supported by OpenGL

Chapter 17: Hardcopy
  - Introduction
  - Definitions
    - Digital images
    - Print
    - Film
    - Video
    - Digital video
    - 3D object prototyping
    - The STL file
  - A word to the wise

Appendices
  - Appendix I: PDB file format
  - Appendix II: CTL file format
  - Appendix III: STL file format

References and Resources
  - References
  - Resources

Evaluation
  - Instructor’s evaluation
  - Student’s evaluation
Because this is a draft of a textbook for an introductory, API-based computer graphics course, the author recognizes that there may be some inaccuracies, incompleteness, or clumsiness in the presentation and apologizes for these in advance. Further development of these materials, as well as source code for many projects and additional examples, is ongoing continuously. All such materials will be posted as they are ready on the author’s Web site:

http://www.cs.csustan.edu/~rsc/NSF/

Your comments and suggestions will be very helpful in making these materials as useful as possible and are solicited; please contact

Steve Cunningham
California State University Stanislaus
rsc@cs.csustan.edu

This work was supported by National Science Foundation grant DUE-9950121. All opinions, findings, conclusions, and recommendations in this work are those of the author and do not necessarily reflect the views of the National Science Foundation. The author also gratefully acknowledges sabbatical support from California State University Stanislaus and thanks the San Diego Supercomputer Center, most particularly Dr. Michael J. Bailey, for hosting this work and for providing significant assistance with both visualization and science content. Ken Brown, a student of the author’s, provided invaluable and much-appreciated assistance with several figures and concepts in this manuscript. The author also thanks students Ben Eadington, Jordan Maynard, and Virginia Muncy for their contributions through examples, and a number of others for valuable conversations and suggestions on these notes.