These notes cover topics in 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, as well as giving students experience in computer graphics programming using the OpenGL application programming interface (API). It also includes discussions of visual communication and of computer graphics in the sciences.

The contents below represent a relatively early draft of these notes. Most of the elements of these contents are in place with the first version of the notes, but not quite all; the contents in this form will give the reader the concept of a fuller organization of the material. Additional changes in the elements and the contents should be expected with later releases.

CONTENTS:

0. Getting Started
   • What is a graphics API?
   • Overview of the notes
   • What is computer graphics?
   • The 3D Graphics Pipeline
     - 3D model coordinate systems
     - 3D world coordinate system
     - 3D eye coordinate system
     - 2D eye coordinates
     - 2D screen coordinates
     - Overall viewing process
     - Different implementation, same result
     - Summary of viewing advantages
   • A basic OpenGL program
   • Mathematics background needed for graphics
     - Coordinate systems and points
     - Line segments and curves
     - Dot and cross products
     - Planes and half-spaces
     - Polygons and convexity
     - Corresponding points in rectangles
     - Line intersections
     - Polar, cylindrical, and spherical coordinates
     - Higher dimensions?

1. Viewing and Projection
   • Introduction
   • Fundamental model of viewing
   • Definitions
     - Setting up the viewing environment
     - Projections
     - Defining the window and viewport
     - What this means
   • 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
- Stereo viewing

2. 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
     - Data structures to hold objects
     - General sources of graphic objects
     - A word to the wise

Transformations and modeling
   • Introduction
   • Definitions
     - Transformation
     - Composite transformation
     - Transformation stacks and their manipulation
   • Compiling geometry

Scene graphs and modeling graphs
   • Introduction
   • A brief summary of scene graphs
     - An example of modeling with a scene graph
   • The viewing transformation
   • Using the modeling graph for coding
     - Example
     - Using standard objects to create more complex scenes
     - Compiling geometry
   • A word to the wise

Implementing modeling in OpenGL
   • The OpenGL model for specifying geometry
     - Point and points mode
     - Line segments
     - Line strips
     - Triangle
     - Sequence of triangles
     - Quads
     - Quad strips
     - General polygon
- 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
  • Transformations in OpenGL
  • Code examples for transformations
    - Simple transformations
    - Transformation stacks
    - Creating display lists

3. Color and Blending
  • Introduction
  • Definitions
    - The RGB cube
    - Luminance
    - Other color models
    - Color depth
    - Color gamut
    - Color blending with the alpha channel
    - Enabling blending
    - Modeling transparency with blending
    - Challenges in blending
  • Some examples
    - An object with partially transparent faces
  • A word to the wise
  • Code examples
    - A model with parts having a full spectrum of colors
    - The HLS double cone
    - An object with partially transparent faces
  • Example code
  • Science projects

4. Visual Communication
  • Introduction
  • Definitions
  • Some examples
    - Different ways encode information
    - Different color encodings for information
    - Geometric encoding of information
    - Other encodings
    - Higher dimensions
    - Choosing an appropriate view
    - Moving a viewpoint
    - Setting a particular viewpoint
    - Legends to help communicate your encodings
    - Implementing legends and labels in OpenGL
  • A word to the wise

5. Science Examples I
  - Modeling diffusion of a quantity in a region
> Temperature in a metal bar
> Temperature in a metal bar
> Spread of disease in a region
- Simple graph of a function of two variables
- Mathematical functions
> Electrostatic potential function
- Simulating a scientific process
> Gas laws
> Diffusion through a semipermeable membrane

6. The OpenGL Pipeline
   • Introduction
   • The Pipeline
   • Implementation in Graphics Cards

7. Lights and Lighting
   • Introduction
   • Definitions
     - Ambient, diffuse, and specular light
     - Use of materials
   • Light properties
     - Positional lights
     - Spotlights
     - Attenuation
     - Directional lights
     - Moving lights
   • Lights and materials in OpenGL
     - Specifying and defining lights
     - Defining materials
     - Setting up a scene to use lighting
     - Using GLU quadric objects
     - Lights of all three primary colors applied to a white surface
     - Code for the example
   • A word to the wise
   • Science examples

8. Shading Models
   • Introduction
   • Definitions
     - Flat shading
     - Smooth shading
   • Other shading models
   • Some examples
   • Code examples
   • Science projects

9. Event Handling
   • Introduction
   • Definitions
   • Some examples of events
     - Keypress events
     - Mouse events
     - system events
     - software events
   • Callback registering
• The vocabulary of interaction
• A word to the wise
• 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
• Definitions
  - Menu bars
  - Buttons
  - Radio buttons
  - Text boxes
  - Horizontal sliders
  - Vertical sliders
  - Text labels
• Using the MUI functionality
• Some examples
• A word to the wise

10. Science Examples II
• Examples
  - Displaying scientific objects
    > Simple molecule display
    > Displaying the conic sections
  - Representing a function of two variables
    > Mathematical functions
    > Surfaces for special functions
    > Electrostatic potential function
    > Interacting waves
  - Representing more complicated functions
    > Implicit surfaces
    > Cross-sections of volumes
    > Vector displays
    > Parametric curves
    > Parametric surfaces
  - Illustrating dynamic systems
    > The Lorenz attractor
    > The Sierpinski attractor
• Some enhancements to the displays
  - Stereo pairs

11. Texture Mapping
• Introduction
• Definitions
  - 1D texture maps
  - 2D texture maps
  - 3D texture maps
  - The relation between the color of the object and the color of the texture map
  - Texture mapping and billboards
• Creating a texture map
  - Getting an image as a texture map
- Generating a synthetic texture map
  • Antialiasing in texturing
  • Texture mapping in OpenGL
    - Capturing a texture from the screen
    - Texture environment
    - Texture parameters
    - Getting and defining a texture map
    - Texture coordinate control
    - 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
  - First example
  - Second example
  - Third example
• References

12. Animation
  • Introduction
  • Definitions
  • Keyframe animation
    - 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

13. High-Performance Graphics Techniques and Games Graphics
  • 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
    - LOD
    - Reducing lighting computation
    - Fog
    - Collision detection
  • A word to the wise

14. Object Selection
  • Introduction
  • Definitions
  • Making selection work
• Picking
• A selection example
• A word to the wise

15. Interpolation and Spline Modeling
• Introduction
  - Interpolations
• Interpolations in OpenGL
• Definitions
• Some examples
• A word to the wise

16. Hardcopy
• Introduction
• Definitions
  - Print
  - Film
  - Video
  - 3D object prototyping
  - The STL file

17. Appendices
• Appendix I: PDB file format
• Appendix II: CTL file format
• Appendix III: STL file format

Evaluation
• Instructor’s evaluation
• Student’s evaluation

Because this is an early draft of the notes for an introductory, API-based computer graphics course, the author apologizes for any inaccuracies, incompleteness, or clumsiness in the presentation. 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@eos.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.