The Computer Graphics Course in the Computational Science Curriculum

Steve Cunningham
California State University Stanislaus

Angela Shiflet
Wofford College

Supercomputing 2002
November 18, 2002
Goals of This Workshop

❖ To discuss the role of computer graphics in computational science studies
❖ To give participants experience in the kind of OpenGL programming found in such a course
❖ To describe a beginning computer graphics course that is a good fit for computational science studies
❖ To discuss some points in teaching this course
Computer Graphics in a Computational Science Program

- Reinforces graphical communication in developing an understanding of scientific principles and processes
- Fits well with work in modeling and simulation that are often in the program
- Adds a dimension in understanding and studying scientific concepts
Problem-Solving Skills Supported by Computer Graphics

❖ Root-Bernstein’s problem-solving tools
  ◆ Pattern seeking and forming
  ◆ Analogizing and abstracting
  ◆ Mental visualization
  ◆ Kinesthetic thinking
  ◆ Playacting (interacting with concepts)

❖ These should be reinforced by written, mathematical, and visual communication
Computer Graphics’ Role in Understanding Science

- Data/Simulation
- Geometrizing
- Geometry
- Rendering
- Information and Insight
- Display
This Role is Critical in Learning

❖ As the student makes hypotheses, the display gives visual feedback about whether the hypothesis fits the data or theory
❖ This feedback loop is critical in developing more sophisticated ideas
❖ Effective displays may need animation and interaction
Visualization’s Role

❖ The complex and subtle phenomena of modern science require visualization to understand well

❖ Communicating that understanding needs effective images to show the critical details

❖ Effective images may be simple or complex and sophisticated; static or animated and interactive
How is the Computer Graphics Provided to the Student?

❖ As a set of pre-written demonstrations that the student executes
❖ As a set of high-level tools (MatLab, Mathematica, ...) the student applies to present data or principles
❖ As a graphics API that a student can apply through programming to create unique explorations of problems
How Well Do the Graphics Tools Cover the Opportunities?
A Computer Graphics Course is an Excellent Way to Give Students Graphics Skills

❖ Engage students in learning how to communicate scientific principles at the same time they are learning graphics
❖ Students understand more fully how the graphics are done and what is possible to do with a full graphics API
Let’s Consider some Examples of Student Work
What Do We See in this Work?

❖ Need to understand principles of the science being examined
❖ Need to think about how the principles can be presented to an audience
❖ Need to iterate from simple to more sophisticated presentations as their understanding grows
❖ Need to work out how to animate their work or make it interactive
Let’s See How This is Done: Hands-on Look at Some Science Projects
How Can You Get Computer Graphics into a Computational Science Curriculum?

- Make your own course
- Collaborate with someone else to create a course that meets your students’ needs as well as theirs
  - We believe that computer science can be a good partner, but so can other disciplines
What Kind of Computer Graphics Course Could it Be?

- Practical rather than algorithmic or theoretical orientation
- Orientated towards graphical problem solving and communication
- Student projects should allow a focus on science topics
- This can be a worthwhile course in CS, engineering, or a science division
Issues in Such a Course

❖ Practical graphics and graphical thinking and problem-solving rather than algorithms and techniques
❖ Projects focused on science rather than on games and other high-visibility areas
❖ The availability of high-quality graphics APIs and equipment makes computer graphics accessible and inexpensive
Course Environment

❖ Almost anything — really. All current systems have OpenGL support
  ◆ Windows 98 and NT
  ◆ Macintosh
  ◆ Workstations
  ◆ Graphics accelerators make a **big** difference

❖ My students email me source from any system; I compile and execute it on mine.
Texts for the Course

❖ Published texts that could be used:

❖ Textbook and materials science focus are being developed; manuscript on CD and online. See credits, last slide.
Content of the Course

❖ First part of the course focuses around the graphics pipeline.
  ◆ Creating images using modeling, transformations, specifying a view, and defining a projection.
  ◆ Color is simple and is defined by simply setting each object’s color
  ◆ Simple animation and interaction are introduced
The Graphics Pipeline

- 3D Model Coordinates
  - Modeling Transformations
    - 3D World Coordinates
      - Viewing Transformation, 3D clipping
        - 3D Eye Coordinates
          - Projection
            - 2D Coordinates
Content of the Course

❖ We introduce the scene graph as a way to organize complex scenes, and students use it in simple ways for modeling.

❖ We do not take full advantage of it until later in the course, when it gives us a straightforward way to create models with hierarchical motion.
The (Simplified) Scene Graph

The basic scene graph only contains group nodes, transformation nodes, and shape nodes that define the actual geometry. It is used to write code by working from the top down, with the transform stack pushed going down a branch and popped when coming up. Geometry is only written for the shape nodes.
Content of the Course

❖ Second part of the course focuses on creating more realistic images
  ◆ Colors and different color models let students make more subtle images
  ◆ Lights, lighting, material definitions, and shading let student projects look better
  ◆ New kinds of events are introduced so that students can create more ways to interact with their projects’ images
Content of the Course

❖ Third part of the course covers ways to create more sophisticated models and images
  ◆ Texture mapping lets students create objects with realistic looks
  ◆ Spline curves and surfaces let students create objects that do not look like polygons
  ◆ Object selection lets students identify objects that should be operated upon
Content of the Course

❖ Throughout, the course focuses on graphics concepts and how they are expressed in a graphics API, not the details of how these concepts are implemented for the API

❖ This lets students move quickly to creating useful small applications for their areas of interest
An Example of this Approach

- Many of the operations in polygon-based graphics are based on interpolating values at polygon vertices as the polygon is scan-converted in screen space.
  - Examples include smooth shading, Phong shading, Z-buffering, texture coordinates
- We describe the process of interpolation but do not go into the actual algorithms.
- Students may get these in a later course.
The Students’ Role in the Course

❖ Students perceive a computing course as much by the projects as by the lectures

❖ We must think about the nature of the student work as a key part of our course design
Give the Student Something to Say

- A problem with traditional computer graphics courses is that their projects can be artificial.
- This course emphasizes a sound background in computer graphics concepts, along with experience in applying these concepts with a current graphics API.
- We include discussions of fields where graphics is important and include information on creating effective visual communication.
- This lets the course serve many audiences.
The Application Area

❖ The sciences are used as an source of many examples and the place where students define most of their projects.

❖ This allows the course to serve as a service course for the sciences (if the student has the background) or as part of a computational science emphasis.

❖ Many other focus areas could be chosen, based on the institution and instructor.
Student Projects

- Student projects pick up the focus area of the course (e.g. the sciences)
- Students identify a topic in the focus area that interests them and I work with each student individually to define the presentation of a problem or concept in this area
Project Sequence in the Course

❖ Simple geometry and colors, some motion.
❖ More complex geometry, smooth shading, some simple interaction.
❖ Significant interaction through menus, mouse, or control panel (MUI, GLUI).
❖ More advanced techniques of the student’s choice.
❖ Spline modeling or object selection.
A Few Things about the Projects

❖ None of the projects simply produces a single image. Each involves at least simple motion or interaction.
❖ Everything is 3D from the start.
❖ Projects are discussed in terms of building images that communicate information, not in terms of making pretty pictures.
Some Examples of Student Work

❖ From the course in fall 1998, CSUS

First project

Second project
Some Examples of Student Work

❖ From the course in fall 1998, CSUS

Third project

Fourth project
Some Examples of Student Work

❖ From the course in fall 1998, CSUS

Fifth project
Some Examples of Student Work

- From the course in fall 1999, SDSU

First project

Second project
Some Examples of Student Work

❖ From the course in fall 1999, SDSU
Some Examples of Student Work

❖ From the course in fall 1999, SDSU

Fifth project
Some Examples of Student Work

A non-project “project” from 1999
Some Examples of Student Work

❖ From the course in fall 2000, CSUS

First project

Second project
Some Examples of Student Work

❖ From the course in fall 2000, CSUS

Third project

Fourth project
Some Examples of Student Work

❖ From the course in fall 2000, CSUS

Fifth project
Some Examples of Student Work

❖ From the course in fall 2002, CSUS

First project

Second project
Credits

❖ This work is supported by National Science Foundation grants DUE-9950121 (Cunningham) and DUE-0087979 (Shiflet). All opinions, findings, conclusions, and recommendations are those of the authors and do not necessarily reflect the views of the National Science Foundation.

❖ Thanks to Mike Bailey of the San Diego Supercomputer Center for valuable help and collaboration and to many students for their contributions.
Postscript

❖ The materials for the course described here are on the CD-ROM, and online at http://www.cs.csustan.edu/~rsc/NSF/

❖ We welcome all comments on the workshop materials and presentation. Our emails are

rsc@cs.csustan.edu
shifletab@wofford.edu