Course Overview

Administrative Issues
Modeling
Animation
Rendering
OpenGL Programming
[Angel Ch. 1]
Course Information On-Line

http://www-bcf.usc.edu/~jbarbic/cs480-s13/

- Schedule (slides, readings)
- Assignments (details, due dates)
- Software (libraries, hints)
- Resources (books, tutorials, links)

Blackboard:
- Forum
- Submit assignments
Course slides

http://www-bcf.usc.edu/~jbarbic/cs480-s13/

• Full-color version

• 6-slides-per-page B&W version
  -- good for printing

• Posted in advance of lectures
  -- bring to class & annotate

• Color viewing in Acrobat Reader:
  Disable “Replace Document Colors” in Preferences. Accessibility (if enabled)
About me

Assistant professor in CS

Post-doc at MIT

PhD, Carnegie Mellon University

jnb@usc.edu

Mon 3:35-5:00, SAL 230
Background:
BSc Mathematics
PhD Computer Science

Research interests:
graphics, animation, real-time physics, control, sound, haptics
Prerequisites

• CSCI 102 Data Structures

• Familiarity with calculus and linear algebra

• C/C++ programming skills

• See me if you are missing any and we haven’t discussed it
Textbooks

- **Interactive Computer Graphics**
  A top-down approach with OpenGL, Fifth Edition
  Edward Angel, Addison-Wesley

- **OpenGL Programming Guide ("Red Book")**
  Basic version also available on-line (see Resources)
Grading

• 51% Programming Assignments (3x 17%)
• 19% Midterm (one sheet of notes only, in class)
• 30% Final (open book)
Academic integrity

• No collaboration!

• Do not copy any parts of any of the assignments from anyone

• Do not look at other students' code, papers, assignments or exams

• USC Office of Student Judicial Affairs and Community Standards will be notified
Assignment Policies

• Programming assignments
  - Hand in via Blackboard by end of due date
  - Functionality and features
  - Style and documentation
  - Artistic impression

• 3 late days, usable any time during semester
• Academic integrity policy applied rigorously
Computer Graphics

One of the “core” computer science disciplines:

Algorithms and Theory
Artificial Intelligence
Computer Architecture
Computer Graphics and Visualization
Computer Security
Computer Systems
Databases
Networks
Programming Languages
Software Engineering
Course Overview

**Theory:** Computer graphics disciplines:
- **Modeling:** how to represent objects
- **Animation:** how to control and represent motion
- **Rendering:** how to create images of objects
- **Image Processing:** how to edit images

**Practice:** OpenGL graphics library

**Not** in this course:
- Human-computer interaction
- Graphic design
- DirectX API
Computer Graphics Disciplines

Rendering

Source: Jensen

Animation

Source: Baraff and Witkin

Geometry (Modeling)

Source: Botsch et al.

Image Processing

Source: Durand
Computer Graphics Goals I

- Synthetic images indistinguishable from reality
- Practical, scientifically sound, in real time
Example: Ray Tracing

Barbic, James,
SIGGRAPH 2010

Thurey, Wojtan,
Gross, Turk,
SIGGRAPH 2010
Example: Physics + Computational Geometry + Animation + Ray Tracing
Example: Radiosity

Computer Graphics Goals II

- Creating a new reality (not necessarily scientific)
- Practical, aesthetically pleasing, in real time
Example: Illustrating Smooth Surfaces

A. Hertzmann, D. Zorin,
SIGGRAPH 2000

Non-photorealistic rendering (NPR)
Example: Scene Completion

Original

Scene Matches

Input

Output

J. Hays, A. Efros,
SIGGRAPH 2007
SIGGRAPH

- Main computer graphics event in the world
- Once per year
- 30,000 attendees
- Academia, industry
1. Course Overview

- Administrative Issues
- Topics Outline (next)
2. OpenGL Basics

• Primitives and attributes
• Color
• Viewing
• Control functions
• [Angel, Ch. 2]
3. Input and Interaction

- Clients and servers
- Event driven programming
- Text and fonts
- [Angel, Ch. 3]
4. Objects & Transformations

- Linear algebra review
- Coordinate systems and frames
- Rotation, translation, scaling
- Homogeneous coordinates
- OpenGL transformation matrices
- [Angel, Ch. 4]
5. Viewing and Projection

- Orthographic projection
- Perspective projection
- Camera positioning
- Projections in OpenGL
- Hidden surface removal
- [Angel, Ch. 5]
6. Hierarchical Models

- Re-using objects
- Animations
- OpenGL routines
- Parameters and transformations
- [Angel, Ch. 10]
7. Light and Shading

- Light sources
- Ambient, diffuse, and specular reflection
- Normal vectors
- Material properties in OpenGL
- Radiosity
- [Angel, Ch. 6]
8. Curves and Surfaces

- Review of 3D-calculus
- Explicit representations
- Implicit representations
- Parametric curves and surfaces
- Hermite curves and surfaces
- Bezier curves and surfaces
- Splines
- Curves and surfaces in OpenGL
- [Angel, Ch. 12]
9. Rendering

- Clipping
- Bounding boxes
- Hidden-surface removal
- Line drawing
- Scan conversion
- Antialiasing
- [Angel, Ch. 7,8]
10. Textures and Pixels

- Texture mapping
- OpenGL texture primitives
- Bump maps
- Environment maps

- Opacity and blending
- Image filtering
- [Angel, Ch. 8]
11. Ray Tracing

- Basic ray tracing [Angel, Ch. 13]
- Spatial data structures [Angel, Ch. 10]
- Motion Blur
- Soft Shadows
12. Radiosity

• Local vs global illumination model
• Interreflection between surfaces
• Radiosity equation
• Solution methods
• [Angel Ch. 13.4-5]
13. Physically Based Models

- Particle systems
- Spring forces
- Cloth
- Collisions
- Constraints
- Fractals
- [Angel, Ch. 11]
14. Scientific Visualization

- Height fields and contours
- Isosurfaces
- Volume rendering
- Texture mapping of volumes

Earth Mantle Heat Convection
University of Utah
Guest Lecture:
TBA

“Wildcard” Lectures:

• Graphics hardware
• More on animation
• Motion capture
• Virtual reality and interaction
• Special effects in movies
• Video game programming
• Non-photo-realistic rendering
Hot Application Areas

• Special effects
• Feature animation
• PC graphics boards
• Video games
• Visualization (science, architecture, space)
Hot Research Topics

- **Modeling**
  - getting models from the real world
  - multi-resolution

- **Animation**
  - physically based simulation
  - motion capture

- **Rendering:**
  - more realistic: image-based modeling
  - less realistic: impressionist, pen & ink
Acknowledgments

• Jessica Hodgins (CMU)
• Frank Pfenning (CMU)
• Paul Heckbert (Nvidia)