Outline

- Computer graphics background
- About the course

What is Computer Graphics?

- Computer-generated images or sequences of images (i.e., animations, movies)
- The scientific study of techniques and methods for generating such images
- Not simply trying for photorealism!
  - Painterly effects
  - Caricatures

http://www.phy.duke.edu/~socolar/family/corcoran.html
http://www.corcoran.org/exhibitions/travel_results.asp?Exhib_ID=63

Some 3-D Computer Graphics Applications

- Manufacturing design (CAD)
- Movies, TV, commercials
  - Animations
  - Special effects mixed with live footage
- Visual arts
- Video games
- Scientific visualization
- Simulation of natural phenomena

Roger Crawfis, Ohio State Univ.
Course description

A first course in computer graphics covering fundamental concepts and techniques related to 2-D and 3-D transformations (including perspective projection), rasterization, shading, hidden surface elimination, and texture mapping, as well as selected topics in modeling, animation and related data structures and mathematical principles.

Outline of course

• Geometry
• Rasterization
• Shading
• Hidden surface elimination
• Texture mapping
• Modeling
• Animation
• Ray tracing
• Global illumination

Outline of course

How to specify the 3-D positions of the camera and the scene objects and their various parts, how to project these to 2-D image locations, and how to represent transformations of these positions.

Outline of course

How to set individual image pixels corresponding to projected geometric objects such as points, lines, polygons, and more complicated shapes. Anti-aliasing reduces artifacts ("jaggies") caused by finite image resolution.
Outline of course

- Geometry
- Rasterization
- Shading
- Hidden surface elimination
- Texture mapping
  - Modeling
  - Animation
  - Ray tracing
  - Global illumination

How to model light interaction with 3-D surfaces with varying material properties in order to calculate the proper colors perceived by the eye at different image locations.

- Ray tracing
- Global illumination

How to efficiently rasterize only the visible parts of scene objects.

Outline of course

- Geometry
- Rasterization
- Shading
- Hidden surface elimination
- Texture mapping
  - Modeling
  - Animation
  - Ray tracing
  - Global illumination

How to apply “layers” of detail to scene objects to show features, simulate bumps and reflections, or other precomputed shading effects. Procedural texturing is concerned with how some kinds of textures are generated algorithmically.

- Ray tracing
- Global illumination

How to efficiently represent the geometry of scene objects, which may be complex, curved, etc. (CSE 784, CSE682)

Outline of course

- Geometry
- Rasterization
- Shading
- Hidden surface elimination
- Texture mapping
  - Modeling
  - Animation
  - Ray tracing
  - Global illumination

Outline of course

- Geometry
- Rasterization
- Shading
- Hidden surface elimination
- Texture mapping
  - Modeling
  - Animation
  - Ray tracing
  - Global illumination

Brown et al, OSU
Outline of course

• Geometry
• Rasterization
• Shading
• Hidden surface elimination
• Texture mapping
  • Modeling
  • Animation
  • Ray tracing
  • Global illumination

How to render dynamic scenes, as well as how to simulate dynamic phenomena (CSE 682)

How to realistically simulate the movement of rays from light sources through multiple object reflections and refractions on the way to the eye (CSE 681)

Outline of course

• Geometry
• Rasterization
• Shading
• Hidden surface elimination
• Texture mapping
  • Modeling
  • Animation
  • Ray tracing
  • Global illumination

How to realistically simulate inter-reflections of light between multiple sources and object surfaces (CSE 782)

CSE 781 will cover these in more detail with the focus on programmable GPU’s and real-time game engine design.
What will I learn from this course?

• A basic understanding of graphics hardware/software technology – algorithms and jargons
• Learn how to use OpenGL to write 2D/3D drawing programs
• Prepare yourself for advanced graphics topics (CIS 681, 682, 694L, 781, 782, 784, 788.xx)

Prerequisites

• Familiarity with:
  – Basic concepts in linear algebra (e.g., vectors, matrices, matrix multiplication)
  – (Object-oriented) programming
  – One of the three C-languages:
    • C
    • C++
    • C#

Lectures

• MWF, 12:30 pm-1:18 pm
• Lecture slides will be posted on the course page before each lecture.

• Read thru the lecture notes and the assigned readings before class.

• Be prepared to ask questions.

Textbook

• Interactive Computer Graphics, A Top-Down Approach Using OpenGL by Edward Angel

  • Very easy to read!
  • Will help you to understand the lectures and prepare for exams
  • Many OpenGL examples in C
Reference Books

The red book
OpenGL programmer’s Guide

The blue book
OpenGL reference manual (man pages)

Grading

• 5 programming assignments worth 55% of course grade.
  – About 2 weeks each
  – Electronic submission through the CSE submit
  – Up to three days late, but 10% penalty for each day
  – No credit after three days, so finish early.

• Exams
  – Midterm: 15%
  – Final: 15%

• Homework
  – Will probably have about 4 home-works, three for grade and one for extra credit – 5% apiece.

Some Jargon

• Graphics Processor or Graphics Processing Unit (GPU)
• What about it?
  – nVIDIA/ATI Graphics Chips
  – ✔ 32-bit colors, Z/stencil buffer
  – ✔ Advanced Per-pixel lighting
  – ✔ Millions of triangles per second
  …

OpenGL programming

• An industry standard API

• This is NOT just a API course
  – You are expected learn the graphics processing and the theory behind it.
  – You don’t need to implement the low-level rasterization algorithms. Instead, we will use OpenGL.
Lab Examples

• Lab1

Lab Examples

• Lab2:
  - Learn how to create and initialize an OpenGL window.
  - Learn how to draw simple 2D primitives (lines, triangles, polygons etc)
  - Learn how to process events: redraw, GUI elements.
• Lab2 Demo

Image Gallery

• Course web site has many examples from two years ago.
• Also see Dr. Shen’s and Dr. Shareef’s pages.

Lab Environment

• Desired Programming environment
  - C/C++/C#
  - OpenGL graphics library
  - Windows Visual Studio .Net
  - Windows PC with dedicated 3D graphics GPU.

• Class discussion.
OpenGL

• CSE machines
  – All Windows-based PC’s will have OpenGL dll’s.
  – Caldwell 112D has a secret PC lab. The machines in this room have ATI Radeon 9800Pro boards.
  – Several machines were recently upgraded to nVidia 8800’s.
  – You now have access to this room.
• Your own machine (or your room-mates)
• More detailed instructions on course page
  I highly recommend trying to get some test programs to compile before you start on HW1

Staying Informed

• Ask fellow students about anything missed in the previous lectures.
• Check the course web site frequently.
• There is a CSE 581 newsgroup for exchanging ideas and asking questions about home-works and study topics
  – I would prefer that most questions be posed here rather than through direct e-mails to me or the TA
  – I will often post e-mail responses to the newsgroup.

Take home message

• What makes most realistic-looking images/animations look so good is a lot of expensive software, artistic and detailed modeling, and a lot of computing power and time
  – For LOTR “Return of the King”, a “renderwall” of ~3,200 CPUs ran 24/7, with an average render time of hours per frame
• The underlying computer graphics principles are what this course will focus on.

For the next lecture...

• Readings
  – Angel: Chapters 1, 2 and 3
Simple Chart

- Playfair
  - hand drawn
  - Circa 1800

Crafting a Simple Chart or Graph

- What are the key ingredients?
- Is object-oriented programming the best model?
- What classes would you implement?
- Does the computer afford a different model or representation?

Anatomy of a Chart

- Class discussion
Advanced features

- Discontinuity
- Smoothing

Advanced features

- Asymptote
- Curve fitting or regression
- Location of axes
- Legends
- Restricted range