Overview

- Introduction
  - What is computer graphics?
- Applications
  - What is it good for?
- Syllabus
  - What will I learn in this course?
- Coursework
  - How much work will there be?
- Examples

Introduction

- What is computer graphics?
  - Imaging = representing 2D images
  - Modeling = representing 3D objects
  - Rendering = constructing 2D images from 3D models
  - Animation = simulating changes over time

Visual Sciences

- Image Processing
- Computer Vision
- Geometric Modeling
- Rendering

Applications

- Entertainment
- Scientific visualization
- Training
- Education
- E-commerce
- Computer art

Examples

- Jurassic Park (Industrial Light & Magic)
- Quake (Id Software)
- Geri’s Game (Pixar Animation Studios)
Movies

Digression

- George Borshukov, vfx technology supervisor, ESC entertainment (The Matrix)

Luxo Jr

- Pixar Animation Studios, 1986
- Director: John Lasseter

Games

Simulation

Applications

- Entertainment
- Computer-aided design
- Scientific visualization
- Training
- Education
- E-commerce
- Computer art
CAD-CAM & design

Virtual reality

Applications: Virtual Reality

Applications

- Entertainment
- Computer-aided design
- Scientific visualization
- Training
- Education
- E-commerce
- Computer art

Visualization

Medical imaging
Applications

- Entertainment
- Computer-aided design
- Scientific visualization
- Training
- Education
- E-commerce
- Computer art

Applications

- Entertainment
- Computer-aided design
- Scientific visualization
- Training
- Education
- E-commerce
- Computer art

Applications

- Entertainment
- Computer-aided design
- Scientific visualization
- Training
- Education
- E-commerce
- Computer art

Questions?
Syllabus (What will I learn in this course? And what not?)

I. Image processing
II. Rendering
III. Modeling
IV. OpenGL
V. Animation

Image Processing
- Image Representation
  - Sampling
  - Reconstruction
  - Quantization & Aliasing
- Image Processing
  - Filtering
  - Warping
  - Morphing
  - Composition
- Raster Graphics
  - Display devices
  - Color models

Rendering
- 3D Rendering Pipeline
  - Modeling transformations
  - Viewing transformations
  - Hidden surface removal
  - Illumination, shading, and textures
  - Scan conversion, clipping
  - Hierarchical scene graphics
  - OpenGL
- Global illumination
  - Ray tracing
  - Radiosity

The Rendering Pipeline
- Transformations
- Clipping
- Rasterization
- Visibility

Ray Casting
- For every pixel construct a ray from the eye
  - For every object in the scene
    - Find intersection with the ray
    - Keep if closest
Ray Tracing
- Original Ray-traced image by Whitted
- Image computed using the Dali ray tracer by Henrik Wann Jensen
- Environment map by Paul Debevec

Traditional Ray Tracing

Ray Tracing+soft shadows

Ray Tracing+caustics

Global Illumination

Shadows
Image-based Rendering
- Use images as inputs and representation
  - E.g. image-based modeling and photo editing
    Boh, Chen, Dorsey and Durand 2001

Modeling
- Representations of geometry
  - Curves: splines
  - Surfaces: meshes, splines, subdivision
  - Solids: voxels, CSG, BSP
- Procedural modeling
  - Sweeps
  - Fractals
  - Grammars

Textures and shading
- Curved surfaces
- Subdivision surfaces

Animation: Keyframing
- Keyframing
  - Kinematics
  - Articulated figures
- Motion capture
  - Capture
  - Warping
- Dynamics
  - Physically-based simulations
  - Particle systems
- Behaviors
  - Planning, learning, etc.
What is Computer Graphics?

- Computer Graphics deals with the tools that one needs in order to:
  - create mathematical models of 2D and 3D objects (*geometric modeling*)
  - produce images given geometrical models (*rendering*)
  - define/represent time-dependent behavior of objects (*animation*). (*“Geometric Modeling” + “Rendering” in 4D.*)

Geometric Modeling

- From a concept (or a real object) to a geometric model on a computer.
- Example: a sphere can be described as \((x,y,z,\tau)\).
- More complex objects can be constructed from simpler ones.

Mesh Representation
Mesh Representation

<table>
<thead>
<tr>
<th>Triangle list</th>
<th>Vertex list</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x,y,z,c$</td>
<td>$i,j,k$</td>
</tr>
<tr>
<td>$x,y,z,c$</td>
<td>$\ldots$</td>
</tr>
<tr>
<td>$x,y,z,c$</td>
<td>$\ldots$</td>
</tr>
<tr>
<td>$i,j,k$</td>
<td></td>
</tr>
</tbody>
</table>

Ignoring attributes, **topology** is about twice as larger as the **geometry**

Overview

- **Introduction**
  - What is computer graphics?
- **Applications**
  - What is it good for?
- **Syllabus**
  - What will I learn in this course?
- **Coursework**
  - How much work will there be?
- **Examples**

Programming Assignments

- Image space – half toning (10%)
- 3D Rendering - Ray Casting + Shading (10%)
- Image Space Morphing (10%)
- OpenGL (TBD) (10%)

Survival Guide

- **Assignments:** 40%
  - Must be completed individually or in pairs
  - No late policy.
- 2-3 Optional Exercises: 30%
  - 1 hour in class
- Final exam: 60% (or down to 30%)

Collaboration Policy

- **Overview:**
  - Working in pairs?
  - You must write your own code (no credit for other code)
  - You must reference your sources of any ideas/code
- It’s OK to …
  - Talk with other students about ideas, approaches, etc.
  - Get ideas from information in books, web sites, etc.
  - Get “support” code from example programs
  - But, you must reference your sources
- It’s NOT OK to …
  - Share code with another student
  - Use ideas or code acquired from another sources without attribution

Quotes from Student Course Guide

- “Yes, if you haven’t heard about it, it’s called Death Graphics. You won’t believe how much work you do for the course.”
- “This class is really a different experience from all other CS courses. If you have the guts, and you have the skills, and of course an interest in graphics, go for it. If you want to find out what a ‘challenging’ semester means, go for it. Also, count this course as 2 courses when you are planning your schedule for the next semester.”