CSE681
Introduction to Computer Graphics

Wikipedia, today
Course Information

- **Instructor:** Han-Wei Shen
- **Email:** hwshen@cse.ohio-state.edu
- **Class URL:**
  - [http://www.cse.ohio-state.edu/~hwshen/681](http://www.cse.ohio-state.edu/~hwshen/681)
  - Find syllabus, schedule, assignments, etc.
- **Office:** DL789
- **Office Hours:** -MW 11:30 – 12:20 or by appointment
Computer Graphics at OSU

CSE 581 (Au, Wi, Sp)

CSE 681 (Au, Wi)

CSE 682 (Wi)

CSE 781 (Wi)

CSE 782 (Au)

CSE 784 (Sp)

CSE 788s
Computer Graphics at OSU

- CSE581
- CSE681
- CSE682
- CSE781
- CSE782
- CSE784
- CSE788

Grad standing
Prerequisites

- CSE560, 581; or graduate standing
- Experience with C/C++ programming skills
- Comfortable with Linear Algebra - vectors and matrices
- Instructor’s STRONG recommendations
  - Develop or have an enthusiasm about Computer Graphics
  - General computer science background: programming, data structures, etc.
  - Good debugging skills
  - Program from a top-down/modular point of view
Textbooks

- **Primary:**
  - *An Introduction to Ray Tracing*, A. Glassner (ed.), Morgan Kaufman

- **Secondary:**
  - *Realistic Ray Tracing*, Peter Shirley and R. Keith Morley, AK Peters
Grading Plan

- Labs: 50%
- Midterm: 20%
- Final: 30%

- Best Image Contest
  - The winners will receive bonus credits
  - The class will vote for the best image candidates
Programming Expectations

- Start labs early!
- Top-down design/Modular programming skills
- Think first then program …
  - Solve problems on paper then code
  - Write algorithms before coding
- Get basic functionality working first then add features as you go
  - Code, debug, and test each new feature completely before tackling the next one
  - Leave yourself enough time for debugging your code
- Write clean and organized code from the beginning … you are graded on this
Permitted Behavior

- High-level discussion on:
  - Clarification as to what the problem asks
  - Ideas to tackle a problem
  - Understand course material
  - Help from me

Not Permitted Behavior

- Peer-to-peer exchange of tangible (bits, bytes, printed) material.
- Solutions that look too similar, e.g., only “cosmetic” differences distinguish two solutions
- Source code (original or enhanced) from outside the classroom setting MUST be acknowledged and a priori permission should be sought.

All suspected cases with sufficient evidence will be reported to the University without discussion.
What is Computer Graphics?

- Computer graphics is “the creation and manipulation of graphics images by means of computer.” (Marc Berger, 1986.)

- Generate synthetic images that look real!
  - Also imaginary worlds (art)

- Real-time (games)

- Accurate (Science, Engineering, Medicine)

- Ease of use
Application Areas

- Movies
- Games
- CAD-CAM
- Simulation
- Virtual reality
- Visualization
- Medical imaging
Movies
Games
Simulation
CAD-CAM & design
Virtual reality
Visualization

VR LAB

[Images of visualization techniques and datasets]
Medical imaging
A Very Broad
Course Overview …
Topics in Computer Graphics

- Modeling
  - Object representation
- Rendering
  - Simulate the image-forming process
  - Projection Based on the pin-hole camera model (this class)
  - Illumination and texturing
- Interaction
  - Virtual environments, peripherals, large displays
  - Real-Time: Human perception wants 30 frames/sec
- Animation
  - Representing/controlling motion
Topics In This Course

1. Viewing and transformation - Mathematical review
2. Ray tracing geometry and organizing ray tracer
3. Illumination
4. Shadows
5. Refraction & Reflection
6. Texture Mapping
7. Anti-aliasing
8. Distributed Ray Tracing
9. Speeding-up ray tracing
Two Rendering Algorithms (1)

- **Backward Projection**
  - Ubiquitous in the industry
  - Graphics hardware support
  - Z-buffer algorithm
  - Polygon-based
  - Special effects not easy to come by

**Z-Buffer**
- For each triangle
- For each projected pixel
- Project scene to the pixels
Two Rendering Algorithms (2)

- **Forward Projection**
  - Ray Tracing *(This course!)*
  - Photorealism
  - No special hardware support
    - Though, attempts are being made to implement it
  - Easy to handle different kinds of object representations
  - Easy to handle special effects
Ray Casting In Action

- For every pixel
  - construct a ray from the eye
    - For every object in the scene
      - Find intersection with the ray
      - Keep if closest
Ray Casting In Action

- 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 In Action

- Shade (interaction of light and material)
- Secondary rays (shadows, reflection, refraction)
Examples
Ray Tracing Textures

(b) Metropolis light transport with 250 mutations per pixel [the same computation time as (a)].
Shadows

Figure 12. Frame from Luxo Jr.

Figure 13. Shadow maps from Luxo Jr.
Traditional Ray Tracing
Ray Tracing + Soft Shadows
Global Illumination (CSE 782)