CMSC427
Computer Graphics
Staff

Instructor
  Prof. Roger Eastman

Teaching assistant
  Patrick Owen
• Course overview: what and why
• Course organization: how
• Getting started: Drawing curves
Computer graphics
Key topics this semester

• Modeling

• Rendering

• Interaction
Creating models of *objects* and *scenes*

- Shape
- Appearance
- Behavior/animation

Techniques

- By hand
- By algorithm
- By capture

Modeling by hand

• Example: Blender (freeware)
  • [https://www.blender.org](https://www.blender.org)

• Professional tools
  • Artistic (Maya, Lightwave)
  • Engineering (Autocad, Solidworks)
  • General (Sketchup)
  • Free AND easy (Tinkercad)
  • Search for 3D modeling tool
Modeling by procedure

- Creating shape, behavior by algorithm

Sweeping Fractal Lines

Dan Gries

(newly generated pic)

E-on Vue software for procedural environments
Modeling by capture

- Measure values from real world
- 3D scanner for static shapes
  - Structure IO sensor
  - Trnio/Scann3D on phones
  - Will do in this class
- Motion capture for dynamics
Synthesis of 2D image from 3D scene

Input
- Data structure that stores object and scene info (geometry, material properties, lights, camera)

Output
- 2D image (array of pixels)
- Red, Green, Blue values for each pixel
Photorealistic rendering

• Physically based simulation of light, materials, camera. Slow, rendering farms, is constantly evolving. Soft shadows, realistic surfaces.
Interactive rendering

- Real time, realistic but approximate physics. Uses specialized GPUs, standard APIs (OpenGL). Hard shadows, cheats in lighting.
Non-photorealistic rendering

• Stylized, cartoonish, for artor illustration

Beyond 2D rendering …

- Stereo VR rendering
- Haptic feedback
  - virtual objects
  - Ultrasound, Univ. of Bristol
- 3D printing!
Interaction – input

- Broad range of input devices beyond keyboard, mouse
- Event driven programming
The Why?

• What’s your interests?
• What’s your experience?
• Why are you taking this course, and what do you want to get out of it?

• Graphics resume assignment for Thursday
• On Canvas
• Submit on Piazza
Course objectives

1. Write efficient interactive 2D and 3D graphics programs using different graphics systems.
2. Create object and scene shape and behavioral models using algorithmic techniques.
3. Render these models at varying levels of photorealism.
4. Describe and apply mathematical and algorithmic foundations as needed for programming, modeling and rendering.
# Graphics systems (limited list of examples!)

<table>
<thead>
<tr>
<th>3D real time APIs</th>
<th>Languages</th>
<th>Game engines</th>
</tr>
</thead>
<tbody>
<tr>
<td>DirectX</td>
<td>C/C++</td>
<td>Unity</td>
</tr>
<tr>
<td><strong>OpenGL</strong></td>
<td><strong>Java (with JOGL)</strong></td>
<td><strong>Unreal</strong></td>
</tr>
<tr>
<td>Vulkan</td>
<td>Java-Processing</td>
<td><strong>Blender</strong></td>
</tr>
<tr>
<td>GLSL</td>
<td>Javascript</td>
<td>Godot</td>
</tr>
<tr>
<td>WebGL</td>
<td>Python</td>
<td>Horde3D</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rendering engines</th>
<th>Widget sets</th>
<th>Physics engines</th>
</tr>
</thead>
<tbody>
<tr>
<td>POV-Ray</td>
<td>QT with C/C++</td>
<td>Box2D</td>
</tr>
<tr>
<td>Orge</td>
<td><strong>Java</strong></td>
<td>Bullet</td>
</tr>
<tr>
<td>Yafaray</td>
<td>Javascript/HTML</td>
<td>Open Dynamics</td>
</tr>
<tr>
<td>Keyshot</td>
<td>TCL/TK</td>
<td>Chipmunk2D</td>
</tr>
<tr>
<td>Renderman</td>
<td>GLUT</td>
<td></td>
</tr>
</tbody>
</table>

**Moral:** evolving, must learn to learn
Course schedule

• Unit I – Object modeling: curves and surfaces

• Unit II – Basics of rendering and OpenGL

• Unit III - Scene modeling: composite objects and scene, 3D interactivity

• Unit IV – Advanced rendering for realism

• Unit V – Advanced modeling for complex shapes
Course organization

• Lecture
  • CSI 1121       TuTh 3:30-4:45 pm

• Canvas
  • Course material and assignments will be posted here.

• Piazza
  • We will use a class discussion forum for answering lecture and assignment questions.
Instructor

Prof. Roger Eastman ([reastman@umd.edu](mailto:reastman@umd.edu))
Office and hours:
A.V. Williams

Teaching assistant

Patrick Owen ([patowen95@gmail.com](mailto:patowen95@gmail.com))
Office and hours:
A.V. Williams
Textbooks

• Required: None
• Provided: David Mount notes on foundations

• Recommended for projects (trade books):
  • Anton’s OpenGL 4 Tutorials, by Anton Gerdelan, Amazon Digital Services, 2014.
  • OpenGL Shading Language (3rd Edition), by Randi J. Rost (Author), et al., Addison-Wesley Professional, 2009

• Recommended for general material (textbooks):
  • Computer Graphics Programming in OpenGL with Java, V. Scott Gordon and John Clevenger, Mercury, 2017
Web resources

• Khronos group:
  • [https://www.khronos.org](https://www.khronos.org)

• Online web tutorials of quality:
  • [https://learnopengl.com](https://learnopengl.com)  Joey de Vries
  • [http://learningwebgl.com/blog/](http://learningwebgl.com/blog/)

• Additional relevant online sources will be distributed through the semester
Prerequisites

Assume you

- Know Java
- Know OOP and data structures (420)
- Are familiar with some linear algebra

Will review

- Matrix operations

Don’t assume you

- Have programmed graphics before
- Have written interactive programs
Assignments and workload

- **Homework (25%)**
  - Weekly homeworks of varying effort and worth

- **Quizzes and exams (30%)**

- **Projects and labs (45%)**
  - Labs: short, focused programming exercises on particular concepts
  - Projects: more substantial programming efforts
Processing

- Complete open source, freeware graphics system from IDE to language to API
- Designed for artists, other "non-CS" types
  - Ben Fry and Casey Reas @ MIT
- Large ecology of supporting libraries
- Used this semester to sketch ideas
- Can be downloaded, or used online:
  - https://processing.org
  - http://sketchpad.cc