CMSC427
Computer Graphics

Staff

Instructor

Prof. Roger Eastman

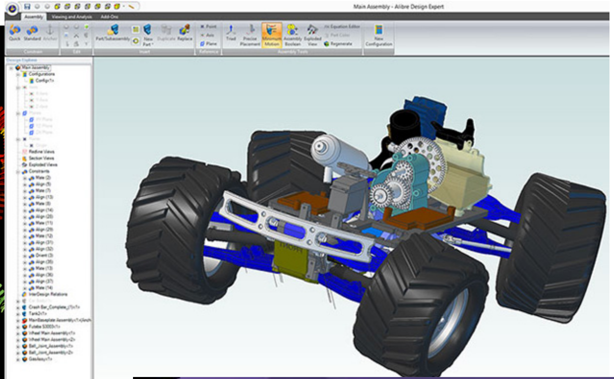
Teaching assistant

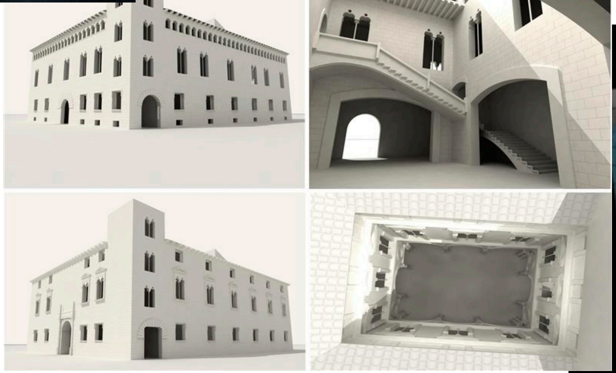
Patrick Owen

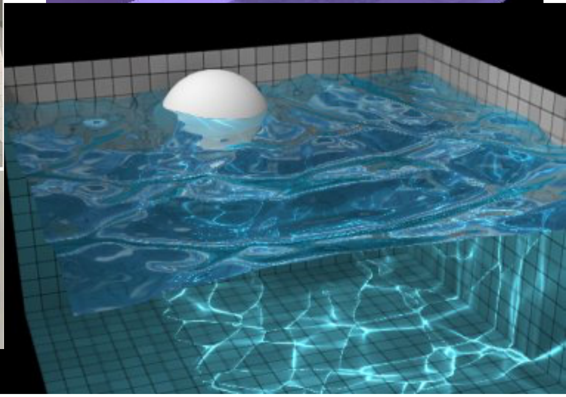
Today

- Course overview: what and why
- Course organization: how
- Getting started: Drawing curves

Computer graphics

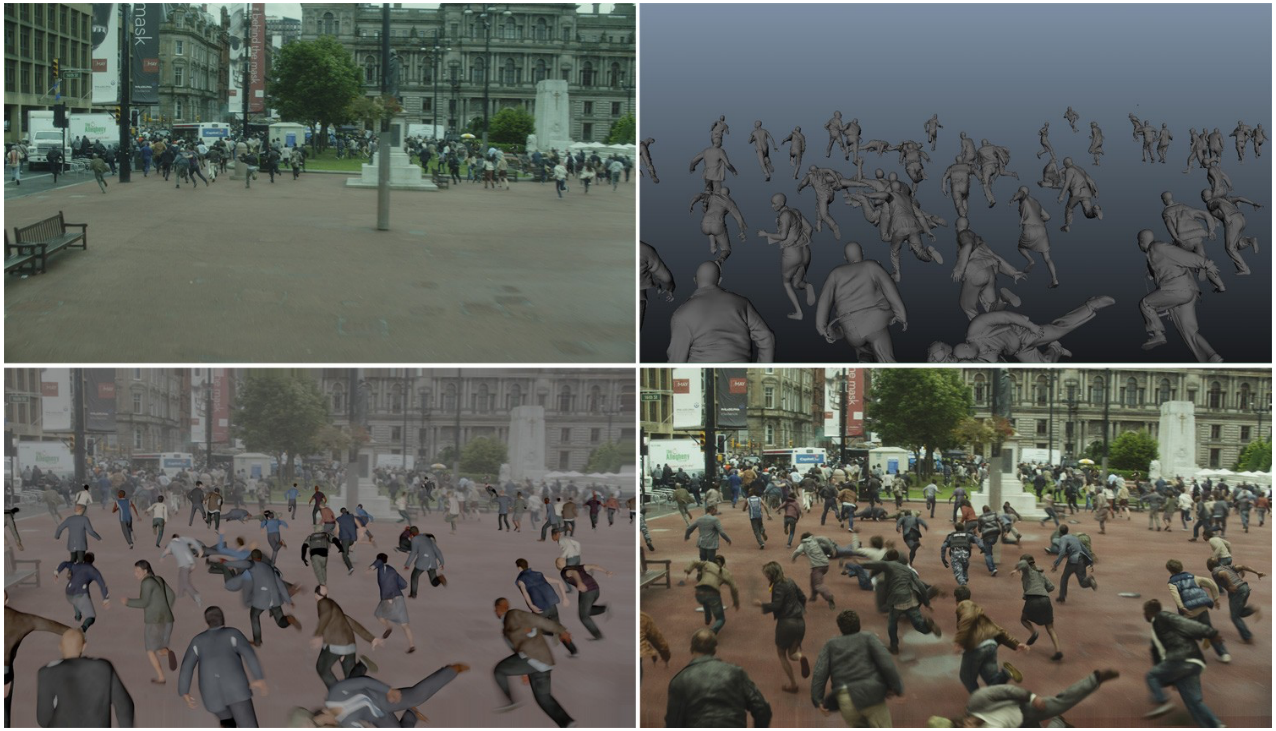




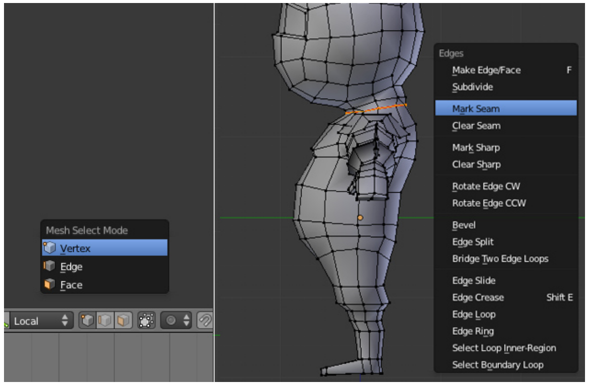


Key topics this semester

- Modeling
- Rendering



- Interaction

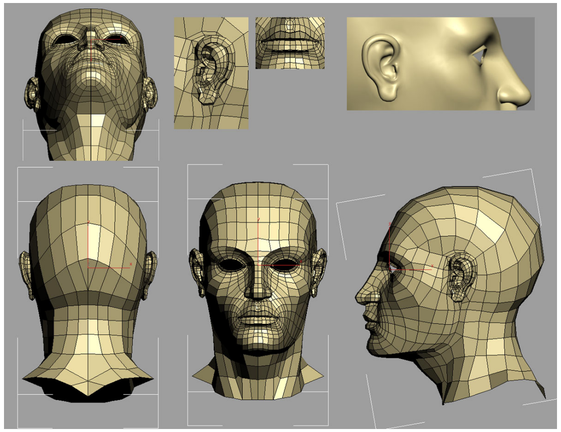


Modeling

- Creating models of *objects* and *scenes*
  - Shape
  - Appearance
  - Behavior/animation

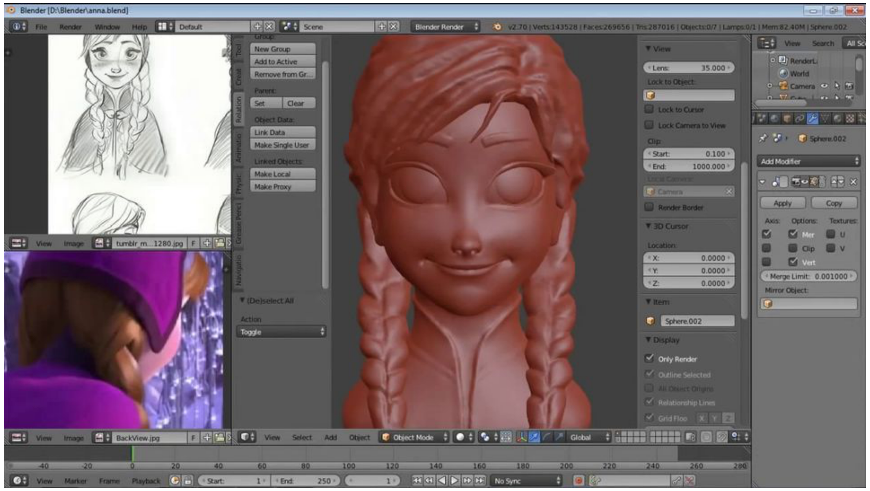
- Techniques
  - By hand
  - By algorithm
  - By capture

http://en.wikipedia.org/wiki/3D_modeling



Modeling by hand

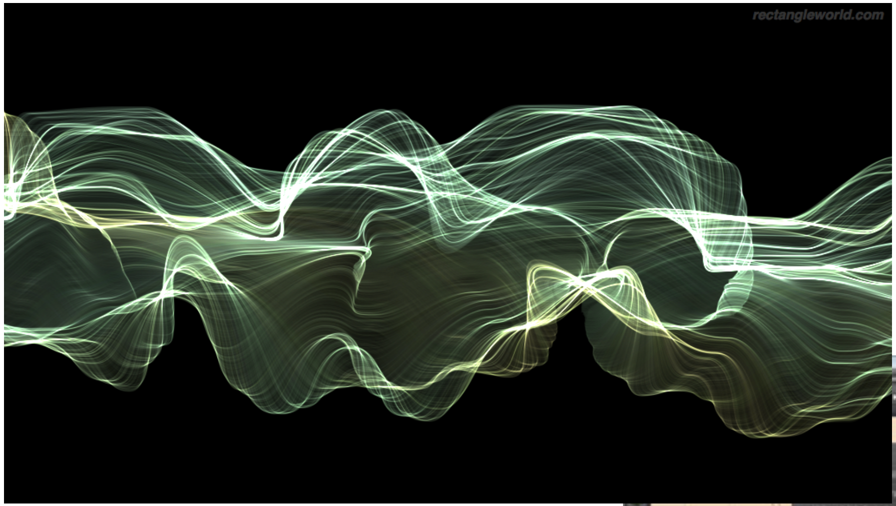
- Example: Blender (freeware)
  - https://www.blender.org



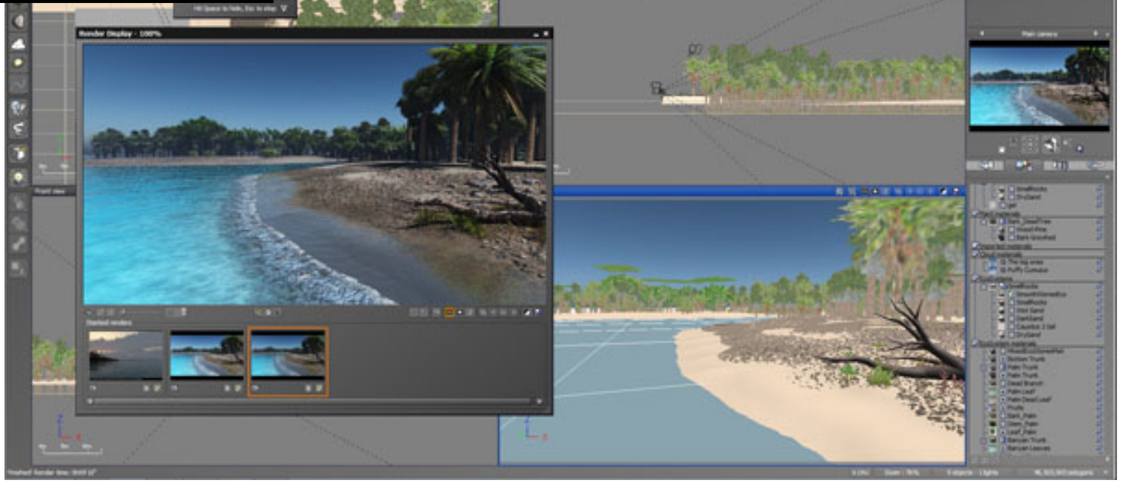
- Professional tools
  - Artistic (Maya, Lightwave)
  - Engineering (Autocad, Solidworks)
  - General (Sketchup)
  - Free AND easy (Tinkercad)
  - Search for 3D modeling tool

Modeling by procedure

- Creating shape, behavior by algorithm



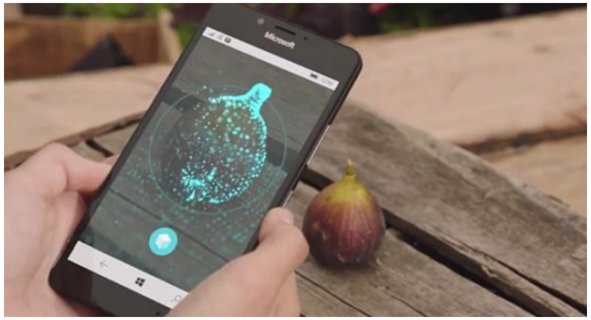
Sweeping Fractal Lines
Dan Gries
(newly generated pic)



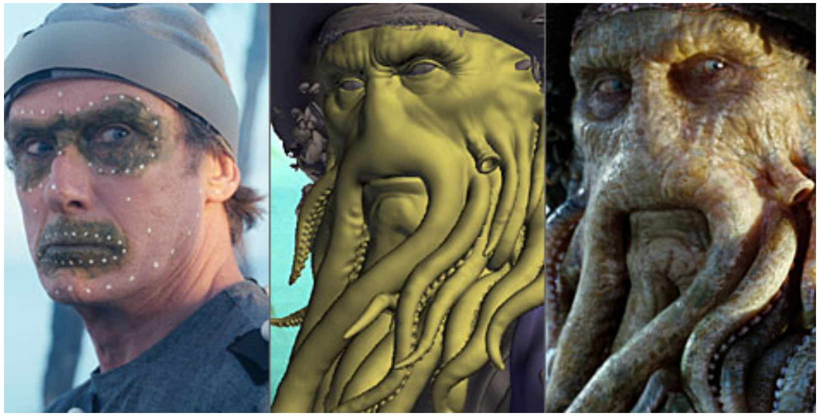
E-on Vue software for procedural environments

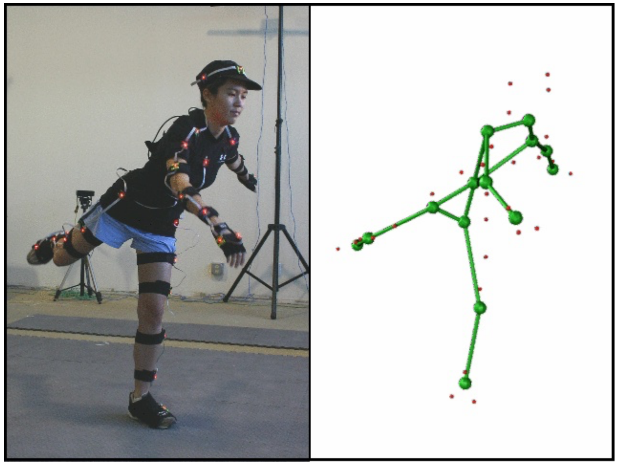
Modeling by capture

- 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





Rendering

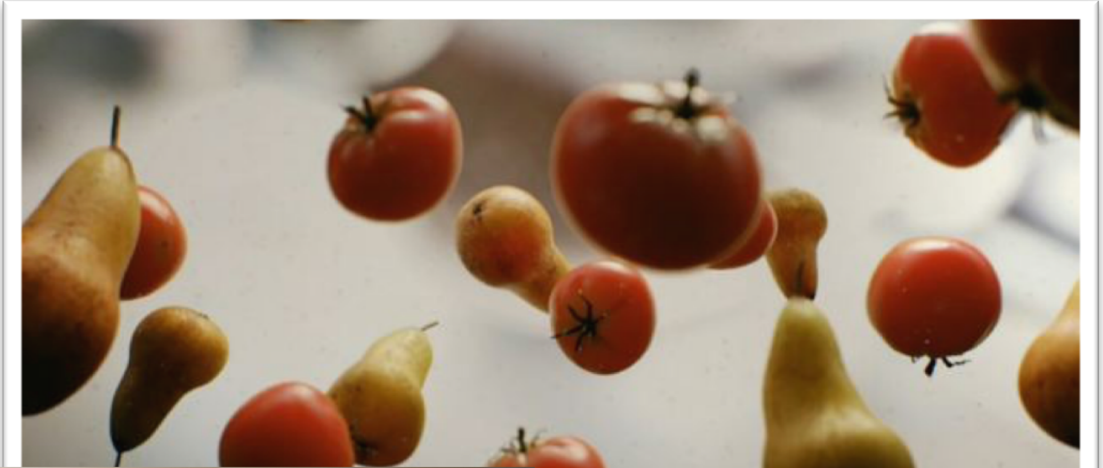
- Synthesis of 2D image from 3D scene
- http://en.wikipedia.org/wiki/Rendering_(computer_graphics)

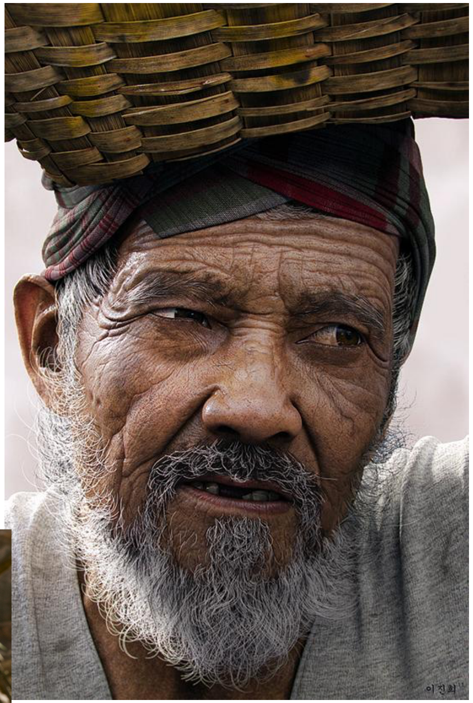
- Input
  - Data structure that stores object and scene info (geometry, material properties, lights, camera)

- 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.

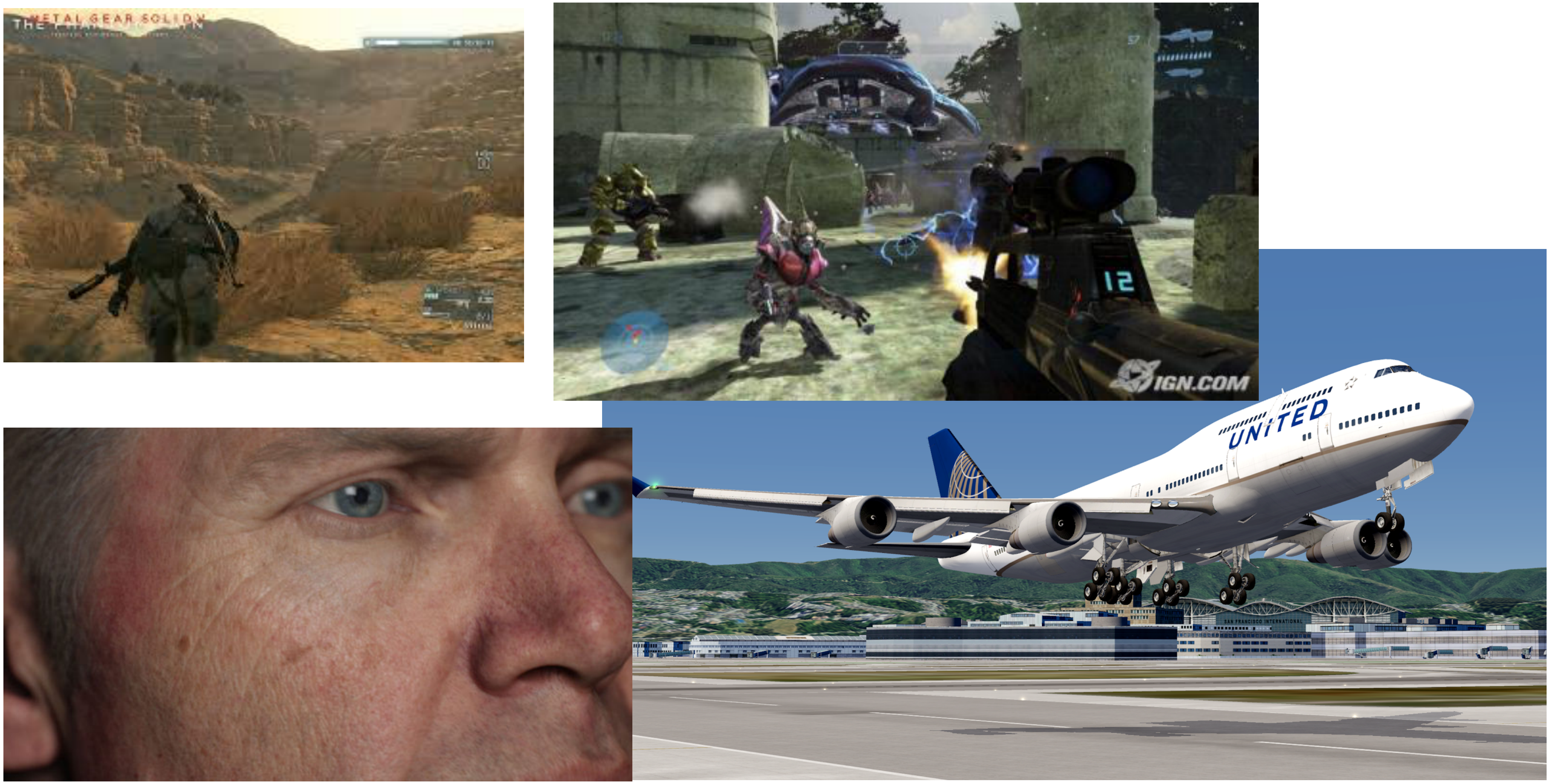




Jin Hee Lee

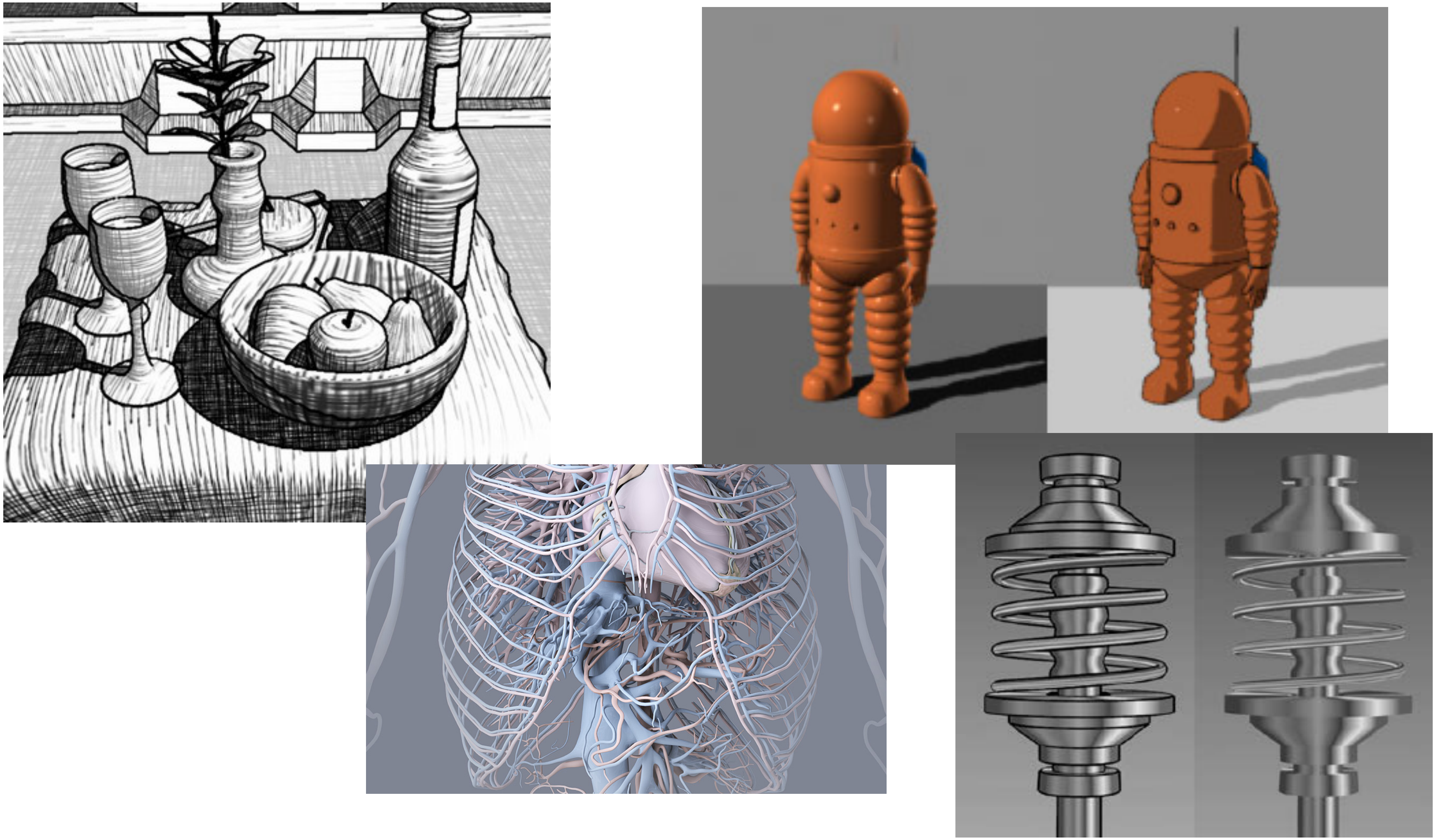
Interactive rendering

- 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
- https://en.wikipedia.org/wiki/Non-photorealistic_rendering



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!)

3D real time APIs

DirectX
OpenGL
Vulcan
GLSL
WebGL

Languages

C/C++
Java (with JOGL)
Java-Processing
Javascript
Python

Game engines

Unity
Unreal
Blender
Godot
Horde3D

Rendering engines

POV-Ray
Orge
Yafaray
Keyshot
Renderman

Widget sets

QT with C/C++
Java
Javascript/HTML
TCL/TK
GLUT

Physics engines

Box2D
Bullet
Open Dynamics
Chipmunk2D

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.

Staff

Instructor

Prof. Roger Eastman (reastman@umd.edu)

Office and hours:

A.V. Williams

Teaching assistant

Patrick Owen (patowen95@gmail.com)

Office and hours:

A.V. Williams

Textbooks

- Required: None
- Provided: David Mount notes on foundations
- Recommended for projects (trade books):
  - OpenGL Superbible (6th edition), by G. Sellers, R. S. Wright, and N. Haemel, 2013.
  - 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
  - Fundamentals of Computer Graphics, Fourth Edition 4th (3rd) Edition, by Steve Marschner and Peter Shirley, AK Peters 2015

Web resources

- Khronos group:
  - https://www.khronos.org
- Online web tutorials of quality:
  - https://learnopengl.com Joey de Vries
  - http://www.iquilezles.org/www/index.htm
  - 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