CMSC427
Computer Graphics
Intro: What and Why

Staff

Instructor

Prof. Roger Eastman

Teaching assistant

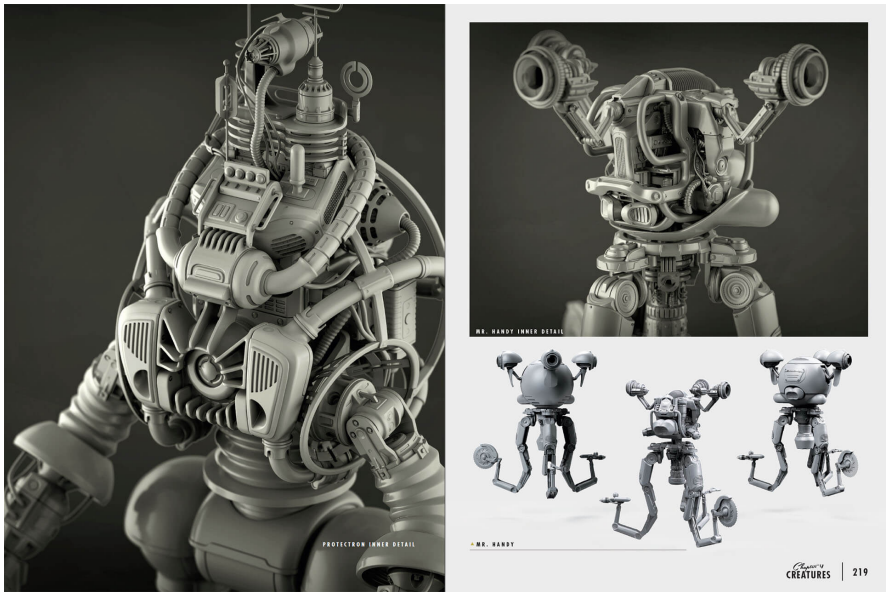
Saeed Hadadan

Former graphics students @ Loyola

Istvan Pely

Game designer

FallOut 3, 4



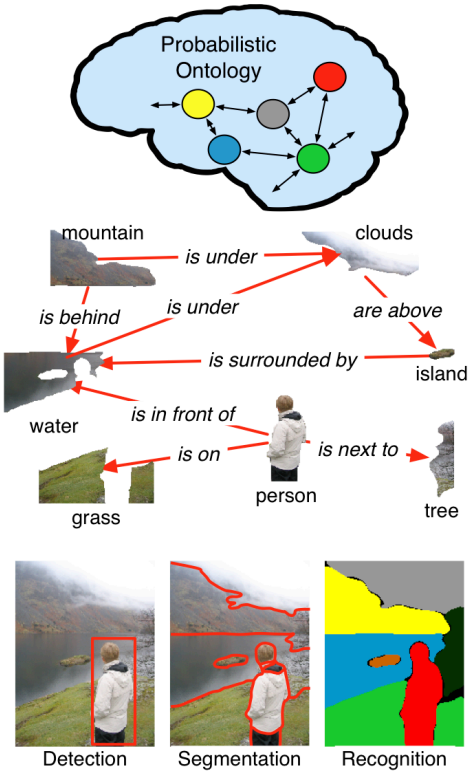
Brian Smith

Film VFX programmer

Matrix, others

Jason Corso

Professor, UMich computer vision



This video

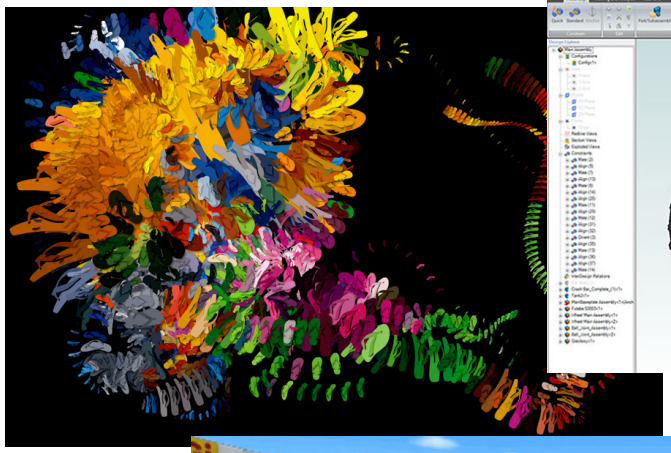
- Course overview: what and why
- Course organization: how

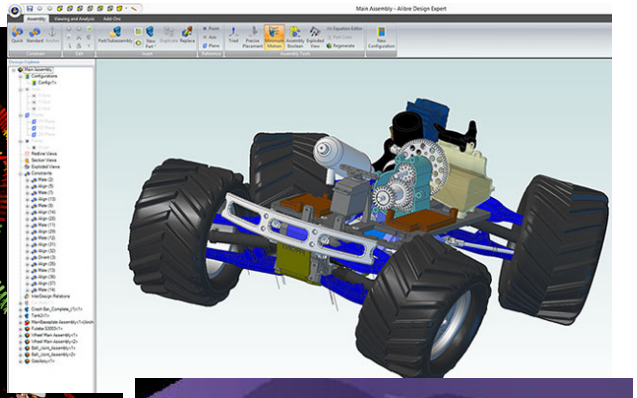
What

- Synthetic imagery

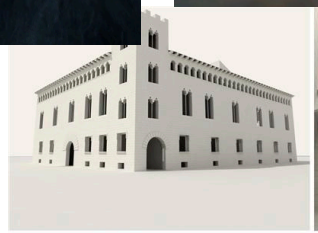
Why

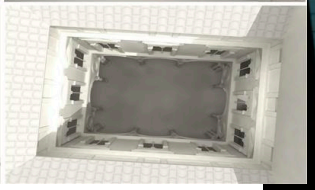
- Computer games
- Special effects in films
- Illustrations
- Art
- Simulations

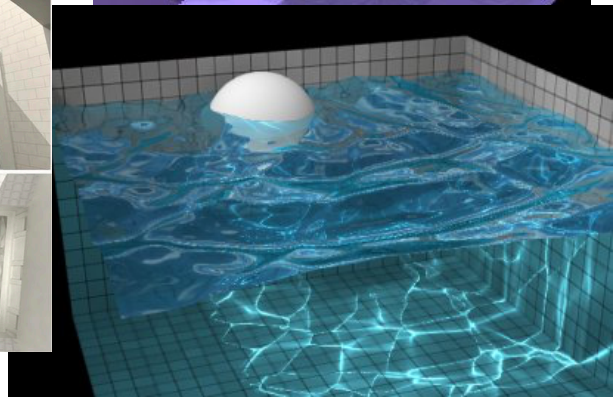





The visible VOXEL-MAN
IMDM University of Hamburg, April 3, 1995



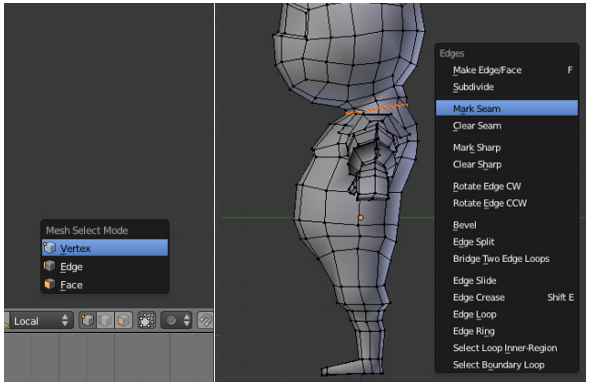




Your interest?

Key topics this semester

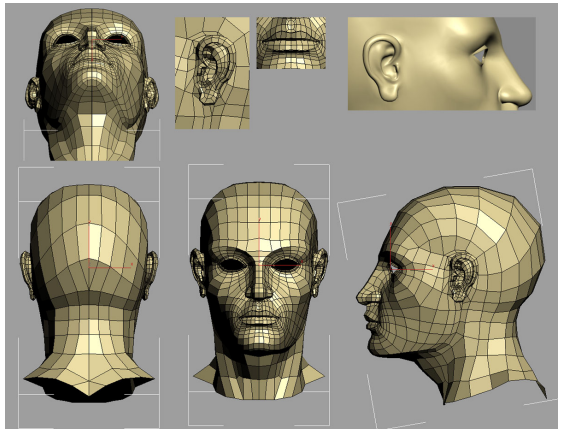
- Modeling
- Rendering
- Interaction



Modeling

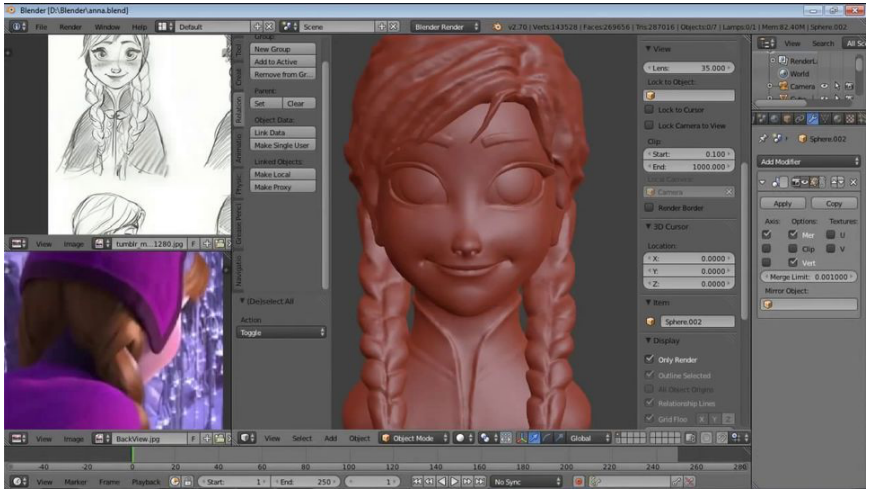
- Creating models of *objects* and *scenes*
  - Shape
  - Appearance
  - Behavior/animation
- Techniques
  - By hand
  - By algorithm
  - By capture

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



Modeling by hand

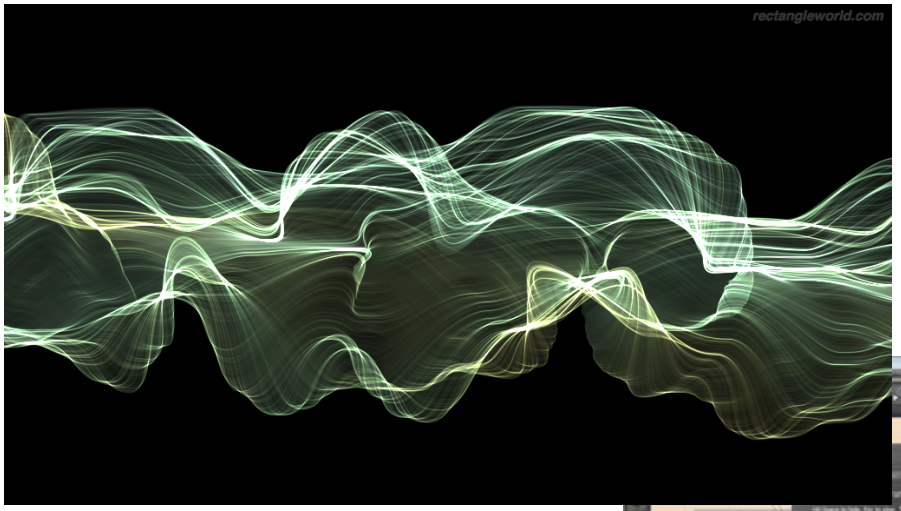
- Example: Blender (freeware)
  - https://www.blender.org



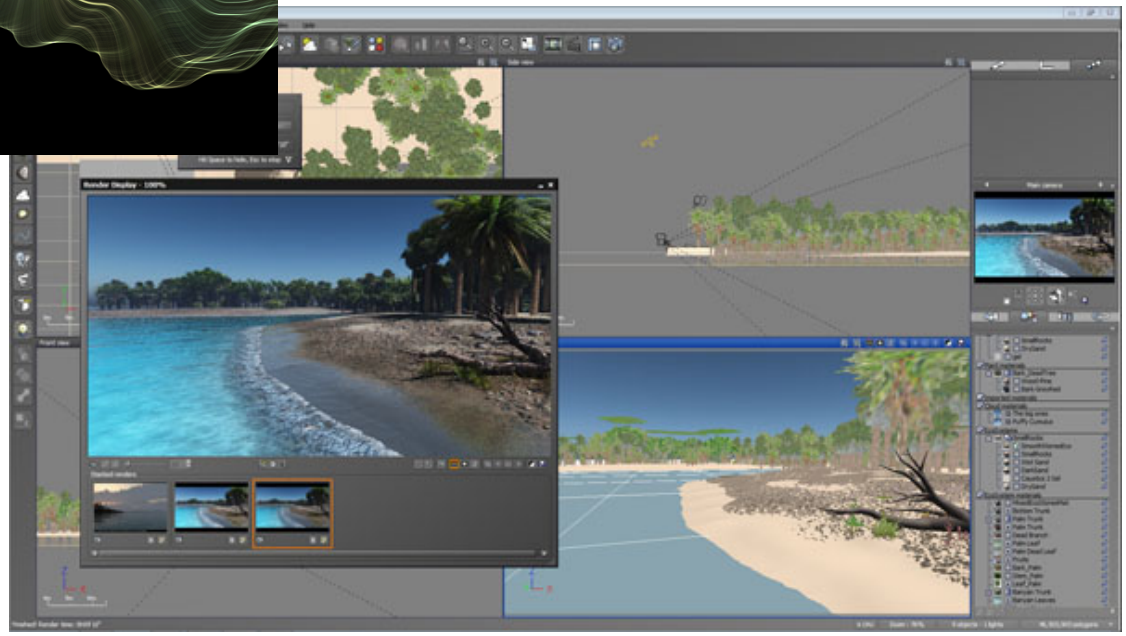
- Professional tools
  - Artistic (Maya, Lightwave)
  - Engineering (Autocad, Solidworks)
  - General (Sketchup)
  - Free AND easy (Tinkercad)
  - Search for 3D modeling tool

Modeling by procedure

- Creating shape, behavior by algorithm



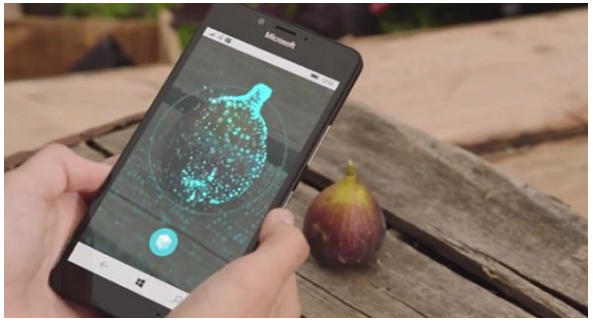
Sweeping Fractal Lines
Dan Gries
(newly generated pic)

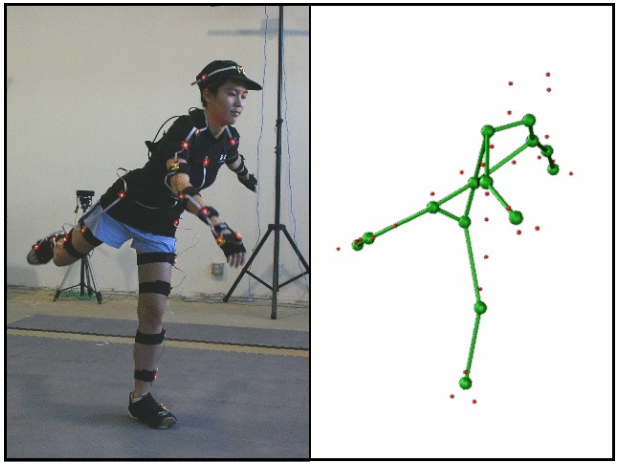


E-on Vue software for procedural environments

Modeling by capture

- Measure values from real world
- 3D scanner for static shapes
  - Structure IO sensor
  - Trnio/Scann3D on phones
- Motion capture for dynamics





Rendering

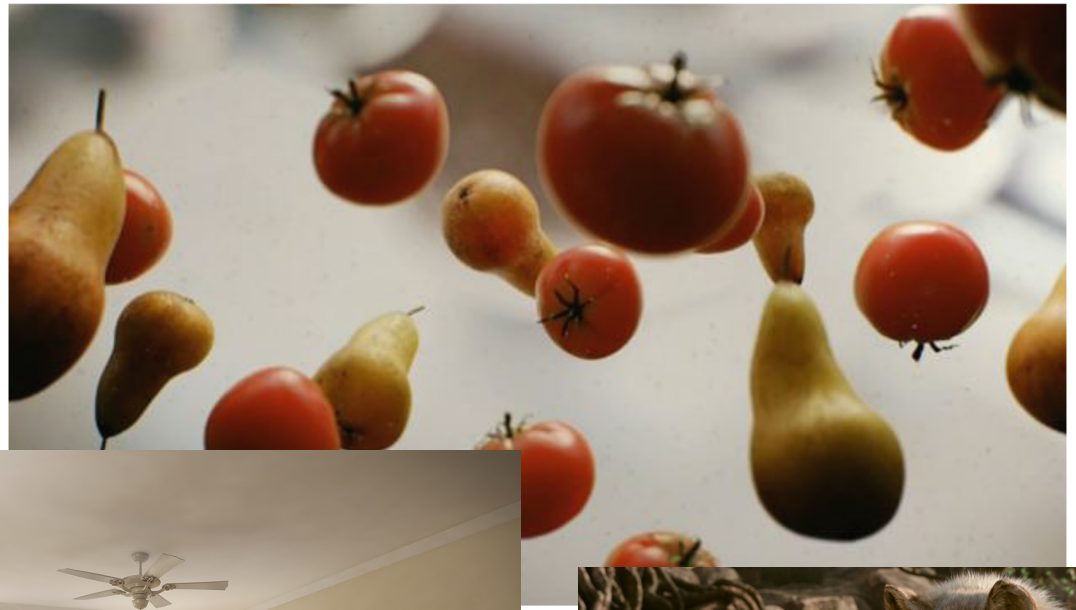
- Synthesis of 2D image from 3D scene
- http://en.wikipedia.org/wiki/Rendering_(computer_graphics)

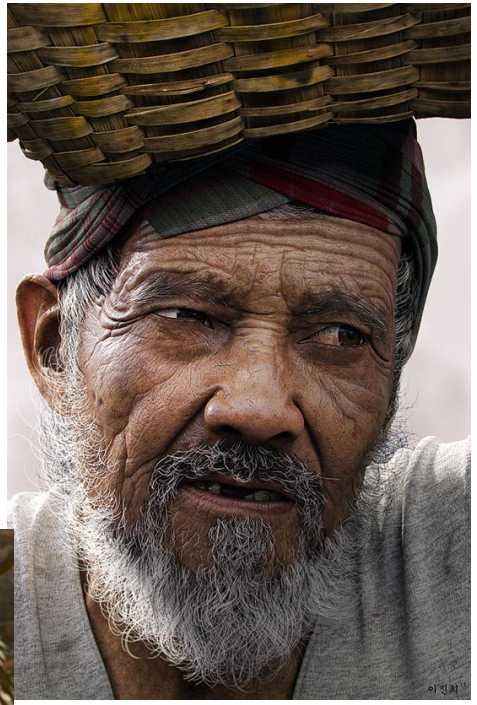
- Input
  - Data structure that stores object and scene info (geometry, material properties, lights, camera)

- 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.





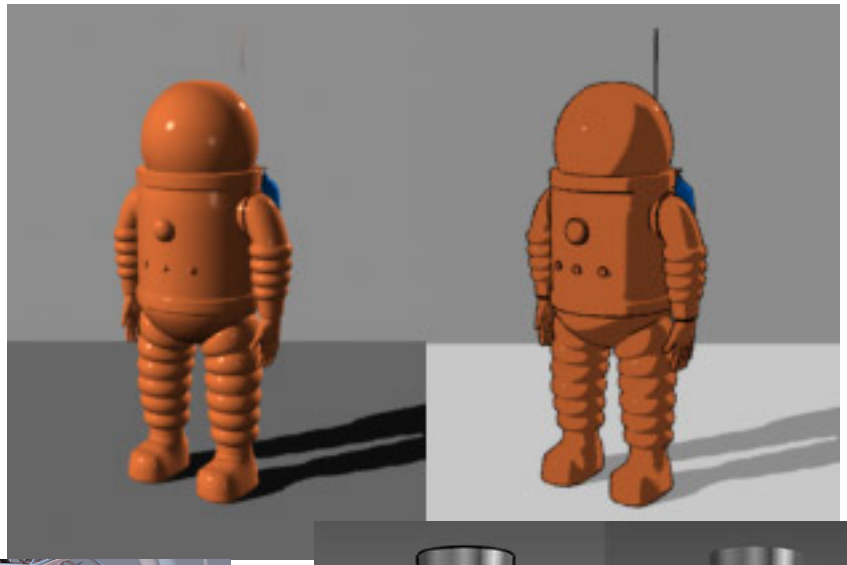
Jin Hee Lee

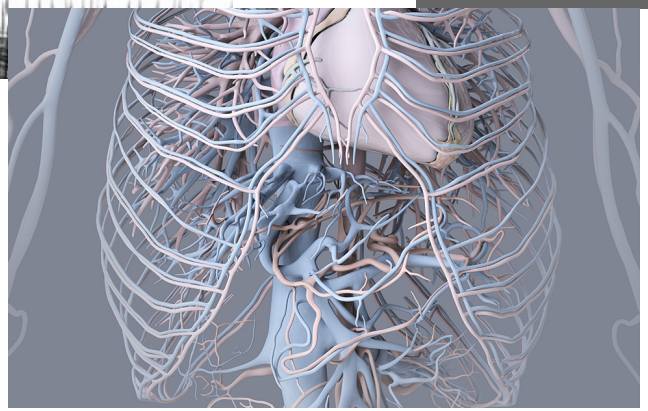
Interactive rendering

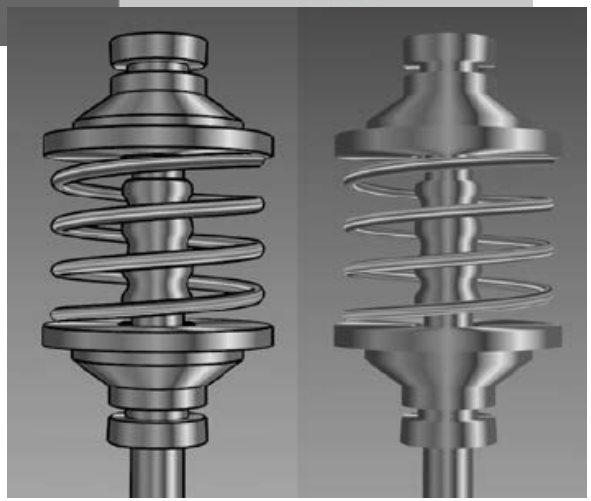
- Real time, realistic but approximate physics. Uses specialized GPUs, standard APIs (OpenGL). Hard shadows, cheats in lighting.

Non-photorealistic rendering

- Stylized, cartoonish, for art or 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 due Thursday Sept. 3rd
- Submit on Canvas

Course objectives

1. Write efficient interactive 2D and 3D graphics programs using different graphics systems.
2. Create object, scene and behavioral models using algorithmic techniques.
3. Render these models at varying levels of realism.
4. Describe and apply mathematical, physical, psychological, and algorithmic foundations as needed for modeling, rendering and interaction.

Course organization

- Lecture
  - Online TuTh 2:00-3:15 pm
  - Will be recorded – attendance not required
  - Will post short videos online before class
  - Class will quickly review material, focus on questions
- Canvas
  - Course material and assignments will be posted here.
- Piazza
  - We will use a class discussion forum for answering lecture and assignment questions.

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