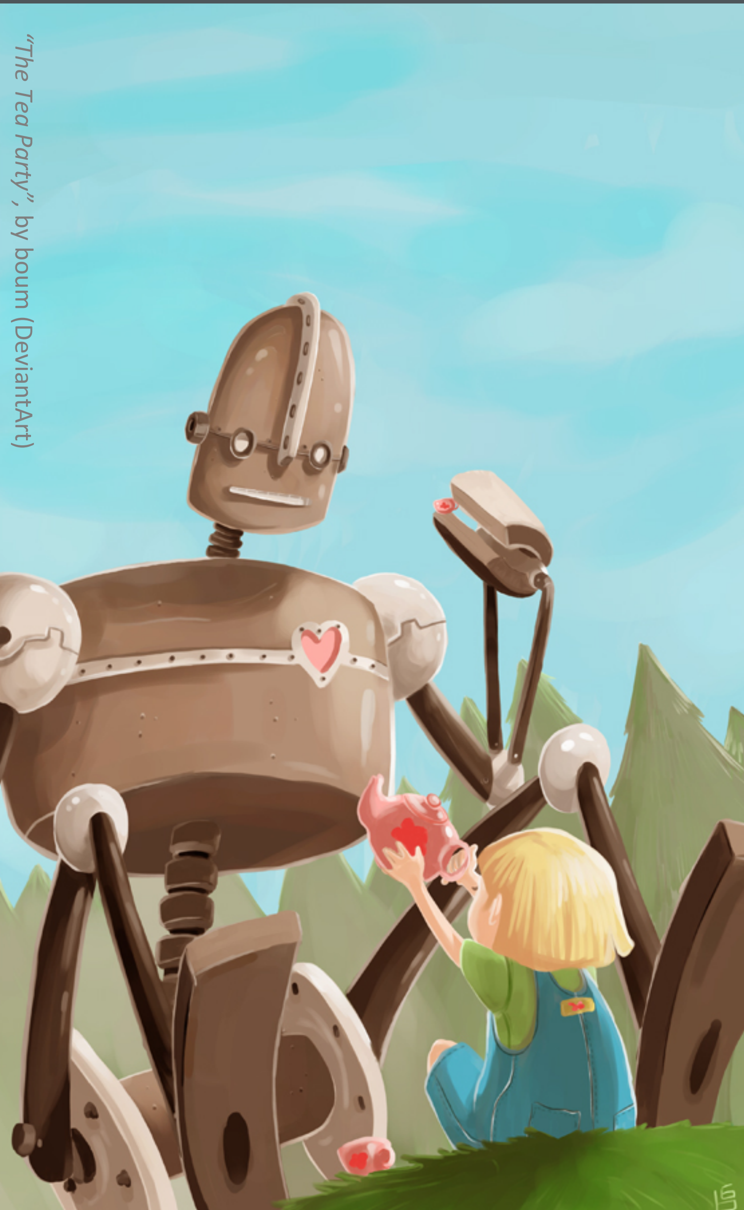
What is Graphics?

course introduction, graphics overview

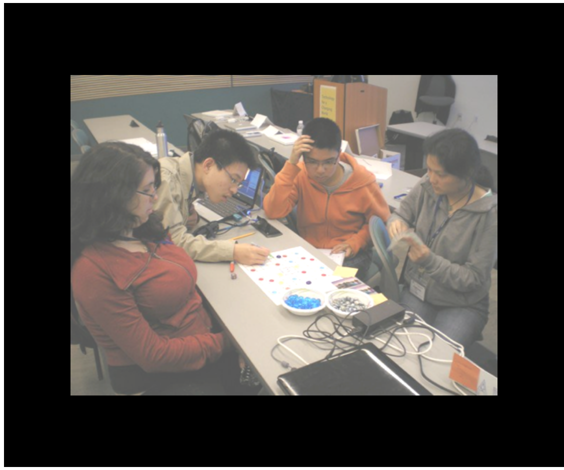
CS 4300/5310
Computer Graphics

COURSE OVERVIEW

A bit about me…



"The Tea Party", by boum (DeviantArt)



ATI Radeon 9700 Tech Demo

Getting Assistance

- Professor: Gillian Smith
- gillian@ccs.neu.edu
- Office hours: Tuesday 1:30 – 3:30pm, WVH 478
  - or by appointment

- TA: Morteza Delgir
- mdelgir@ccs.neu.edu
- Office hours: ?

Getting Assistance

- Professor: Gillian Smith
  - Develop and deliver course lectures
  - Develop assignments and project guidelines
  - Grading presentations, projects
  - Office hours (or appointment) to answer concept questions, assignment/project clarification
  - Checks Piazza regularly
- TA: Morteza Delgir
  - Develop and deliver lab lectures
  - Lab office hours to answer assignment questions, practical questions, coding/math help
  - Grading assignments, reading responses
  - Checks Piazza regularly

Class Overview

Provide a broad foundation in computer graphics **concepts**:

- Color representation
- Rendering pipeline
- Images as 2D signals
- 3D rendering methods
- Graphics topics

Website: http://www.ccs.neu.edu/course/cs5310sp13

Class Overview

Provide a broad foundation in computer graphics **practice**:

- 2D drawing, animation, and image processing
- Raytracing
- 3D with OpenGL
- Programmable graphics hardware: shaders

Website: http://www.ccs.neu.edu/course/cs5310sp13

Class Overview

Reinforce and build transferrable skills:

- Learning new languages and APIs
- Improve written and oral communication skills
- Effective strategies for reading research papers

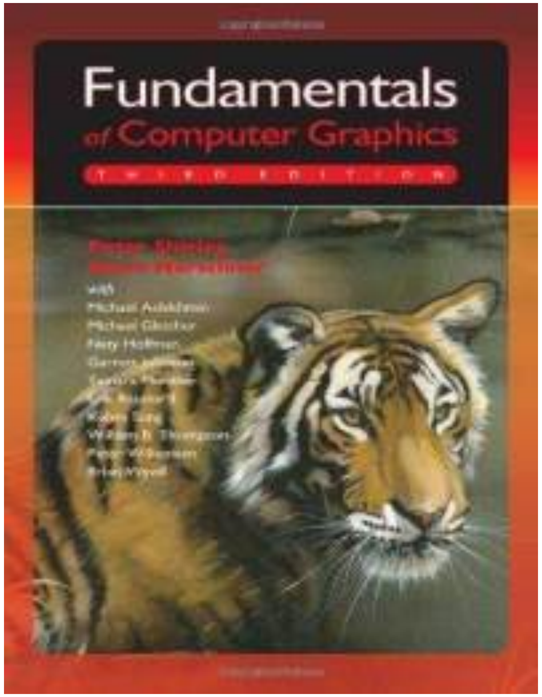
Website: http://www.ccs.neu.edu/course/cs5310sp13

Textbook

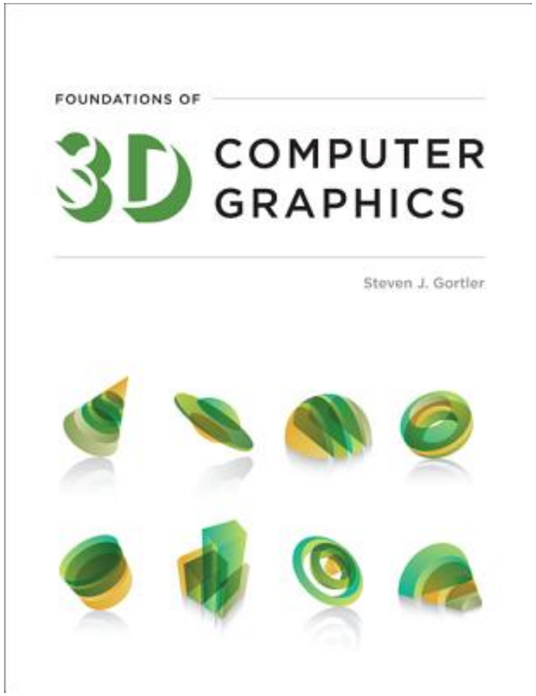
Fundamentals of Computer Graphics

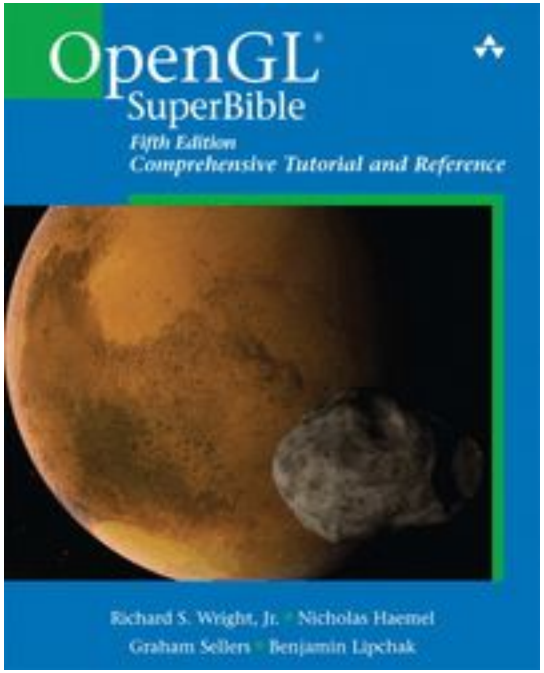
Third Edition

Peter Shirley



Other Useful Resources *(optional)*





Foundations of 3D Computer Graphics
Third Edition

Steven J. Gortler

OpenGL SuperBible
Fifth Edition

Richard S. Wright

Collaboration Policy

- I **encourage** you to…
  - Share ideas with other students
  - Work together to come up with general solutions
  - Discuss papers prior to response
- I **require** you to…
  - Write your own code for assignments
  - Turn in all written assignments in your own words
  - Understand **every** aspect of code you turn in
  - Give detailed credit to people you have worked with or online resources
  - Ask if you have any questions about this policy

Collaboration Policy

- I **encourage** you to…
  - Share ideas with other students
  - Work together to come up with general solutions
  - Discuss papers prior to response
- I **require** you to…
  - Write your own code for assignments
  - Turn in all written assignments in your own words
  - Understand **every** aspect of code you turn in
  - Give detailed credit to people you have worked with or online resources
  - Ask if you have any questions about this policy
- **Plagiarism and copying is strictly forbidden, and will result in disciplinary action.**

Avoiding Plagiarizing and Copying

- **Never** share your code with another student
  - Sharing your code is still considered cheating, even if you wrote it all yourself!
    - No matter how good your intent!
- **Never** copy text or code directly from any source
- **Always** give attribution to external sources
  - Quoting text appropriately, citing to the source
  - Citing paraphrased text
  - Acknowledge collaborators and external sources

Paraphrasing?

- Do not just replace some or all words with synonyms
- Do not keep the same sentence or paragraph structure
- Do not copy bits and pieces of sentences and move them around

- Do?
  - Write it **in your own words**.
  - Add context and clarity to the original material

Grading Overview

- Course Participation: 10%
- Reading Responses: 5%
- Assignments: 30%
- 2D Midterm Project: 20%
- 3D Final Project: 35%

Grading Overview

- 93 – 100 A ☺ (excellent!)
- 90 – 92 A-
- 88 – 89 B+
- 83 – 87 B (good!)
- 80 – 82 B-
- 78 – 79 C+
- 73 – 77 C
- 70 – 72 C-
- 60 – 69 D
- <60 F ☹

Grading Overview

93 – 100	A	☺	(excellent!)
90 – 92	A-		
88 – 89	B+		
83 – 87	B		(good!)
80 – 82	B-		
78 – 79	C+		
73 – 77	C		
70 – 72	C-		
60 – 69	D		
<60	F	☹	

Your grade is your responsibility. If you need a particular grade for any reason, **you** are solely responsible for **earning** it.

Course Participation

- Discussion in-class
- Piazza online discussion forum
  - Link is in the syllabus

Reading Responses

- One page, due at **10am** on the day of class
- Brief, **two sentence** summary of the reading
- The rest should be **your opinion**
  - What are the shortcomings of the work?
  - What would you do to extend it?
  - What is the future work?
  - Is there anything you strongly disagree with? Strongly agree with?
  - How does the paper relate to your interests?

Assignment Grading Policy

- All assignments weighted equally
- Every day an assignment is late, 10% off
- Each of you has **five** late days
  - Split them across assignments as you wish
  - Use for planned absences or unexpected issues
  - For assignments **only**
  - When turning in assignment late, specify how many late days you will use
- Unexpected, longer-term emergencies: see me!

Midterm Projects

- Groups of 2-3 students
  - More or less by my permission only (and have a really good reason)
- Game? Painting program? Image processing?
- Phases:
  - Proposal: January 22
  - Presentation: February 5
  - Paper: February 5

Final Projects

- Groups of 2-3 students
  - Can be different from midterm
- Game? Image synthesis? Reproduce movie scene?
- Phases:
  - Proposal: March 19
  - Status Report: March 28
  - Presentation: April 9, April 11, April 16
  - Report: April 16

Project Grading

- Midterm Project
  - Proposal: 40%
  - Presentation: 20%
  - Project: 40%

- Final Project
  - Proposal: 20%
  - Status Report: 10%
  - Presentation: 30%
  - Project: 40%

Art Contest!

- Accidental Art
  - "oops" moments that turn out to be interesting
- Intentional Art
  - Intentionally created, interesting and/or beautiful art

- To enter:
  - Upload screenshot or video (link to youtube) on Blackboard
  - One entry per contest per student per assignment
  - Two entries per contest per group per project

- We vote on winner at end of semester

Programming Environment

- Processing
  - 2D graphics
  - 3D with OpenGL (mostly)
  - Shader support
  - Recommended and supported
  - In-class examples

- Also permitted
  - Java and JOGL (Java2D for 2D assignment/project)
  - C++ and OpenGL
  - Javascript and WebGL

- **All homework turned in must run on CCIS lab machines**

Choosing an Environment: Processing

- Pros
  - Rapid prototyping tool
  - Simple to start with
  - Very powerful
- Awareness
  - Limited debugging support
- Cons
  - **Not** what professional games companies use
  - Does not always support advanced OpenGL

Choosing an Environment: C++ & OpenGL

- Pros
  - More advanced
  - Closer to what game companies use
- Cons
  - More advanced
  - More development environment setup
  - Overkill for 2D

Choosing an Environment: Javascript/WebGL

- Pros
  - New and exciting
  - Runs in almost any browser
  - Forefront of web development technology
- Cons
  - New = less existing support
  - Less instructor support

How do I choose?

- What are your goals?
  - Professional
  - Educational
- Getting hands dirty
  - Pro: much more control
  - Con: can be harder
- Prior graphics programming experience

Final Class Thoughts

- You get out of a class what you put into it!
- I promise
  - Lectures teaching concepts
  - Interesting additional readings
  - Relevant assignments to reinforce concepts
  - Project guidance
- You promise
  - Asking and answering questions (in class, online)
  - Participation in discussion
  - Diligent, timely work on assignments
  - Independent learning

Questions?

WHAT IS GRAPHICS?

What is Graphics?

GRAPHICS

What is Graphics?

Math + Programming + Art

What is Graphics?

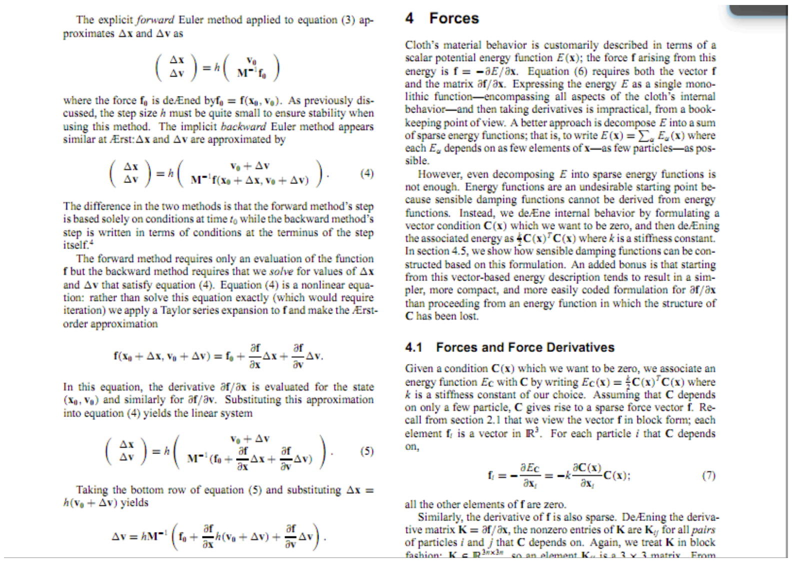
Math + Programming **+ Art**

uh oh!

What is Graphics?

uh oh?

Math + Programming + Art



The explicit *forward* Euler method applied to equation (3) approximates $\Delta\mathbf{x}$ and $\Delta\mathbf{v}$ as

$$\begin{pmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{v} \end{pmatrix} = h \begin{pmatrix} \mathbf{v}_0 \\ \mathbf{M}^{-1}\mathbf{f}_0 \end{pmatrix}$$

where the force $\mathbf{f}_0$ is deÆned by $\mathbf{f}_0 = \mathbf{f}(\mathbf{x}_0, \mathbf{v}_0)$. As previously discussed, the step size h must be quite small to ensure stability when using this method. The implicit *backward* Euler method appears similar at Ærst: $\Delta\mathbf{x}$ and $\Delta\mathbf{v}$ are approximated by

$$\begin{pmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{v} \end{pmatrix} = h \begin{pmatrix} \mathbf{v}_0 + \Delta\mathbf{v} \\ \mathbf{M}^{-1}\mathbf{f}(\mathbf{x}_0 + \Delta\mathbf{x}, \mathbf{v}_0 + \Delta\mathbf{v}) \end{pmatrix}. \quad (4)$$

The difference in the two methods is that the forward method's step is based solely on conditions at time t_0 while the backward method's step is written in terms of conditions at the terminus of the step itself.[4]

The forward method requires only an evaluation of the function $\mathbf{f}$ but the backward method requires that we *solve* for values of $\Delta\mathbf{x}$ and $\Delta\mathbf{v}$ that satisfy equation (4). Equation (4) is a nonlinear equation: rather than solve this equation exactly (which would require iteration) we apply a Taylor series expansion to $\mathbf{f}$ and make the Ærst-order approximation

$$\mathbf{f}(\mathbf{x}_0 + \Delta\mathbf{x}, \mathbf{v}_0 + \Delta\mathbf{v}) = \mathbf{f}_0 + \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\Delta\mathbf{x} + \frac{\partial \mathbf{f}}{\partial \mathbf{v}}\Delta\mathbf{v}.$$

In this equation, the derivative $\partial\mathbf{f}/\partial\mathbf{x}$ is evaluated for the state $(\mathbf{x}_0, \mathbf{v}_0)$ and similarly for $\partial\mathbf{f}/\partial\mathbf{v}$. Substituting this approximation into equation (4) yields the linear system

$$\begin{pmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{v} \end{pmatrix} = h \begin{pmatrix} \mathbf{v}_0 + \Delta\mathbf{v} \\ \mathbf{M}^{-1}(\mathbf{f}_0 + \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\Delta\mathbf{x} + \frac{\partial \mathbf{f}}{\partial \mathbf{v}}\Delta\mathbf{v}) \end{pmatrix}. \quad (5)$$

Taking the bottom row of equation (5) and substituting $\Delta\mathbf{x} = h(\mathbf{v}_0 + \Delta\mathbf{v})$ yields

$$\Delta\mathbf{v} = h\mathbf{M}^{-1}\left(\mathbf{f}_0 + \frac{\partial \mathbf{f}}{\partial \mathbf{x}}h(\mathbf{v}_0 + \Delta\mathbf{v}) + \frac{\partial \mathbf{f}}{\partial \mathbf{v}}\Delta\mathbf{v}\right).$$

4 Forces

Cloth's material behavior is customarily described in terms of a scalar potential energy function $E(\mathbf{x})$; the force $\mathbf{f}$ arising from this energy is $\mathbf{f} = -\partial E/\partial\mathbf{x}$. Equation (6) requires both the vector $\mathbf{f}$ and the matrix $\partial\mathbf{f}/\partial\mathbf{x}$. Expressing the energy E as a single monolithic function—encompassing all aspects of the cloth's internal behavior—and then taking derivatives is impractical, from a bookkeeping point of view. A better approach is decompose E into a sum of sparse energy functions; that is, to write $E(\mathbf{x}) = \sum_\alpha E_\alpha(\mathbf{x})$ where each E_α depends on as few elements of $\mathbf{x}$—as few particles—as possible.

However, even decomposing E into sparse energy functions is not enough. Energy functions are an undesirable starting point because sensible damping functions cannot be derived from energy functions. Instead, we deÆne internal behavior by formulating a vector condition $\mathbf{C}(\mathbf{x})$ which we want to be zero, and then deÆning the associated energy as $\frac{k}{2}\mathbf{C}(\mathbf{x})^T\mathbf{C}(\mathbf{x})$ where k is a stiffness constant. In section 4.5, we show how sensible damping functions can be constructed based on this formulation. An added bonus is that starting from this vector-based energy description tends to result in a simpler, more compact, and more easily coded formulation for $\partial\mathbf{f}/\partial\mathbf{x}$ than proceeding from an energy function in which the structure of $\mathbf{C}$ has been lost.

4.1 Forces and Force Derivatives

Given a condition $\mathbf{C}(\mathbf{x})$ which we want to be zero, we associate an energy function $E_\mathbf{C}$ with $\mathbf{C}$ by writing $E_\mathbf{C}(\mathbf{x}) = \frac{k}{2}\mathbf{C}(\mathbf{x})^T\mathbf{C}(\mathbf{x})$ where k is a stiffness constant of our choice. Assuming that $\mathbf{C}$ depends on only a few particle, $\mathbf{C}$ gives rise to a sparse force vector $\mathbf{f}$. Recall from section 2.1 that we view the vector $\mathbf{f}$ in block form; each element $\mathbf{f}_i$ is a vector in $\mathbb{R}^3$. For each particle i that $\mathbf{C}$ depends on,

$$\mathbf{f}_i = -\frac{\partial E_\mathbf{C}}{\partial \mathbf{x}_i} = -k\frac{\partial \mathbf{C}(\mathbf{x})}{\partial \mathbf{x}_i}\mathbf{C}(\mathbf{x}); \quad (7)$$

all the other elements of $\mathbf{f}$ are zero.

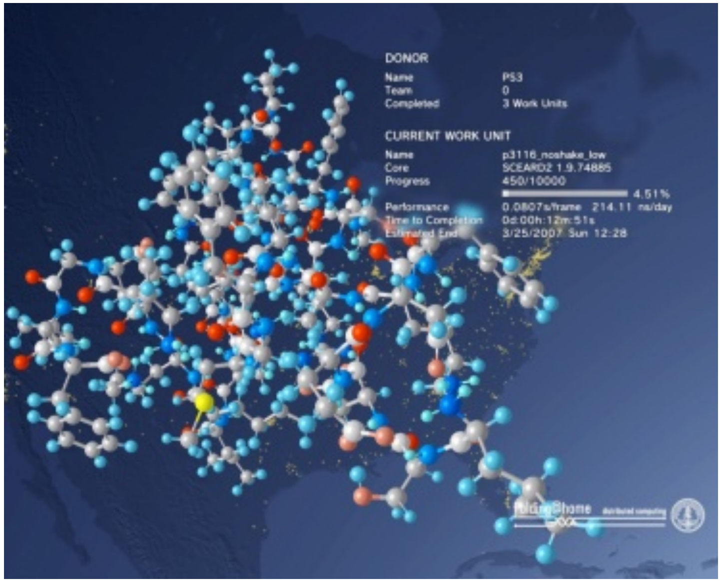
Similarly, the derivative of $\mathbf{f}$ is also sparse. DeÆning the derivative matrix $\mathbf{K} = \partial\mathbf{f}/\partial\mathbf{x}$, the nonzero entries of $\mathbf{K}$ are $\mathbf{K}_{ij}$ for all *pairs* of particles i and j that $\mathbf{C}$ depends on. Again, we treat $\mathbf{K}$ in block fashion: $\mathbf{K} \in \mathbb{R}^{3n\times 3n}$, so an element $\mathbf{K}_{ij}$ is a 3×3 matrix. From

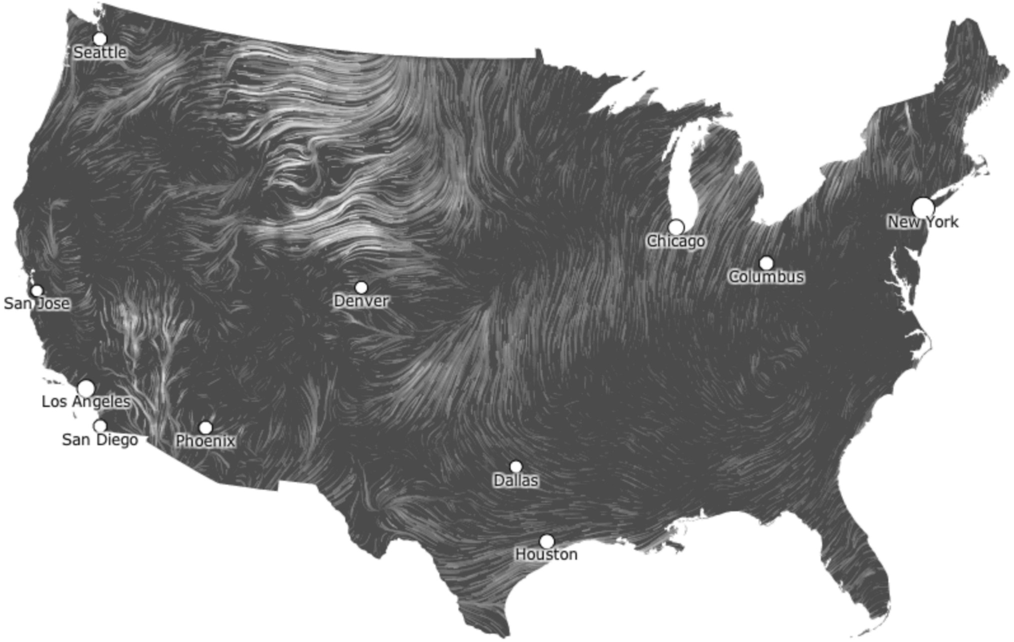
Application Areas: Film Production

Application Areas: Special Effects

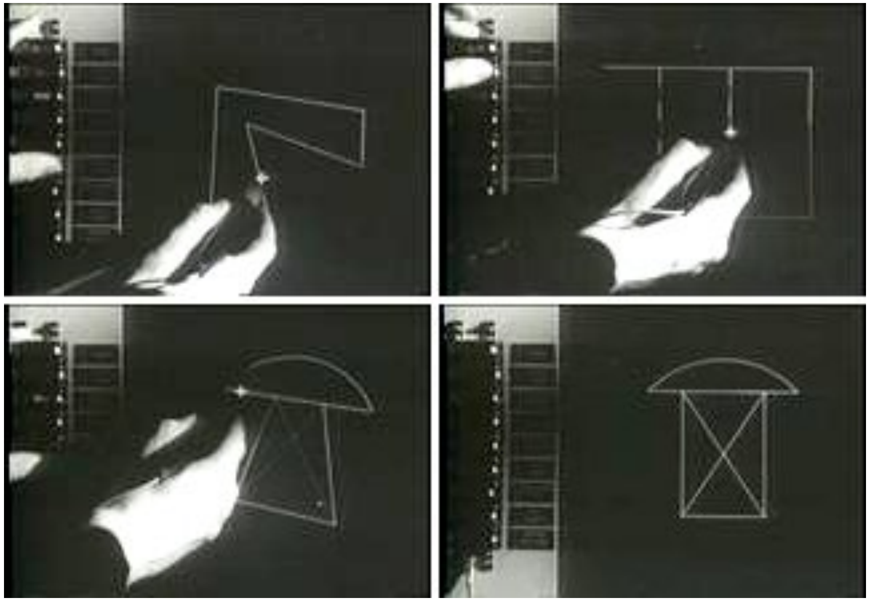
Application Areas: Games

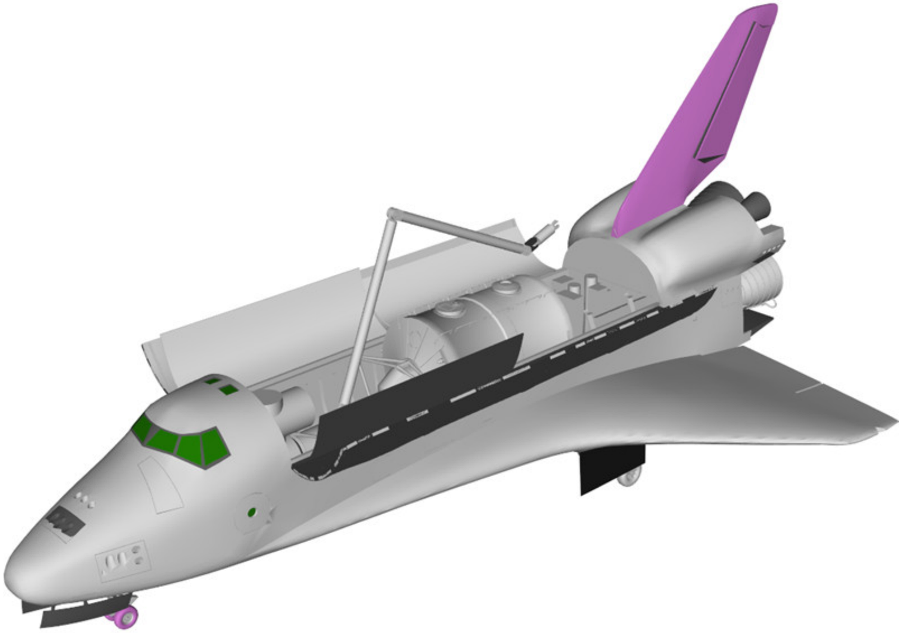
Application Areas: Scientific Visualization





Application Areas: CAD





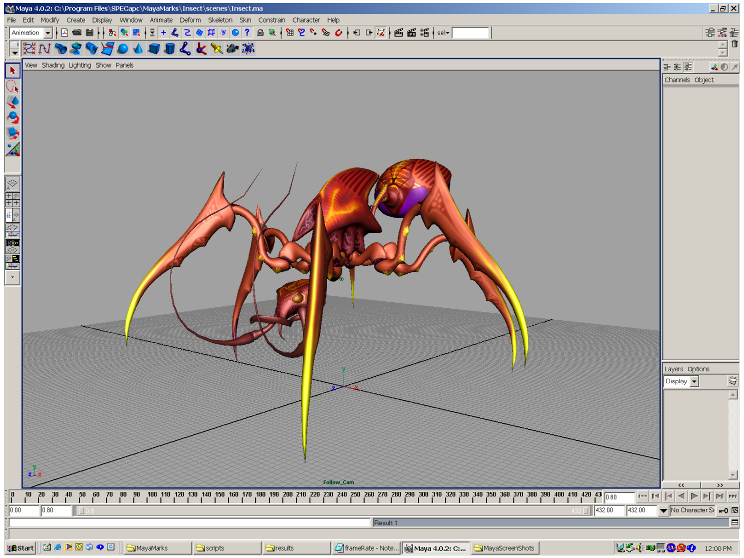
Application Areas: Simulation

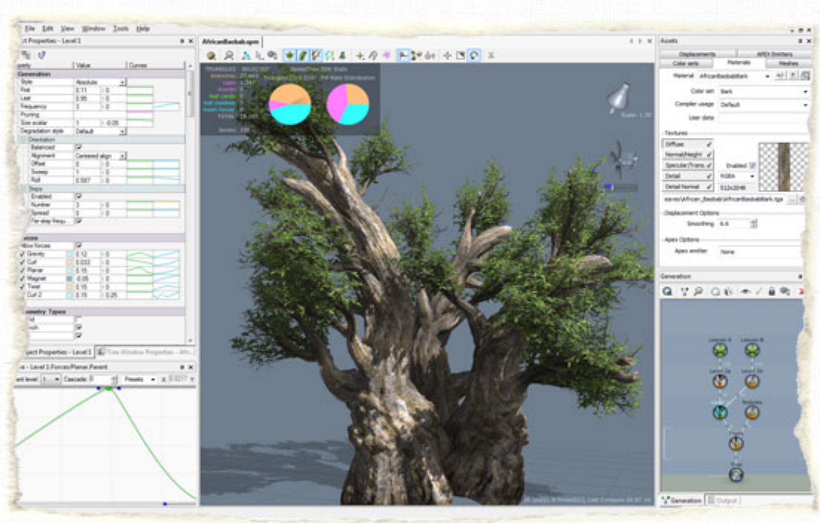
Application Areas: Illustration Tools

Application Areas: Fine Arts

Graphics Problems

- **Modeling**
  - 2D and 3D representation
  - Curved surfaces
  - Procedural techniques
- Rendering
  - Realism
  - Speed
  - Non-realism
- Animation
  - Illusion of life
  - Motion capture
  - Keyframing
  - Physical simulation



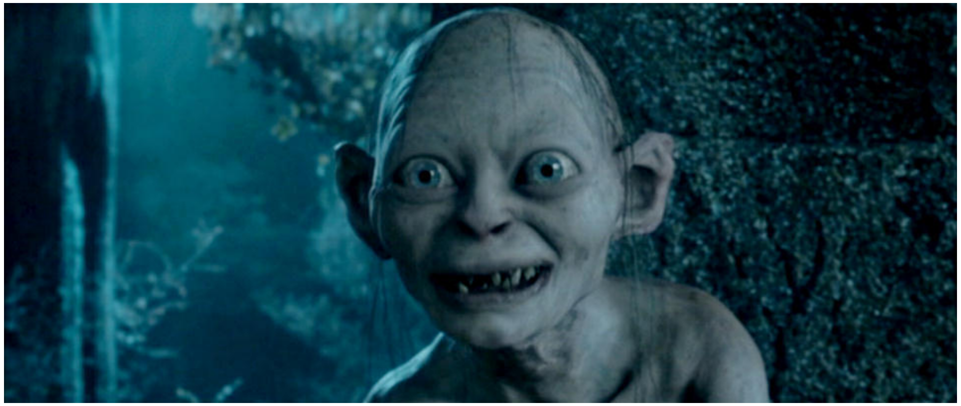


Graphics Problems

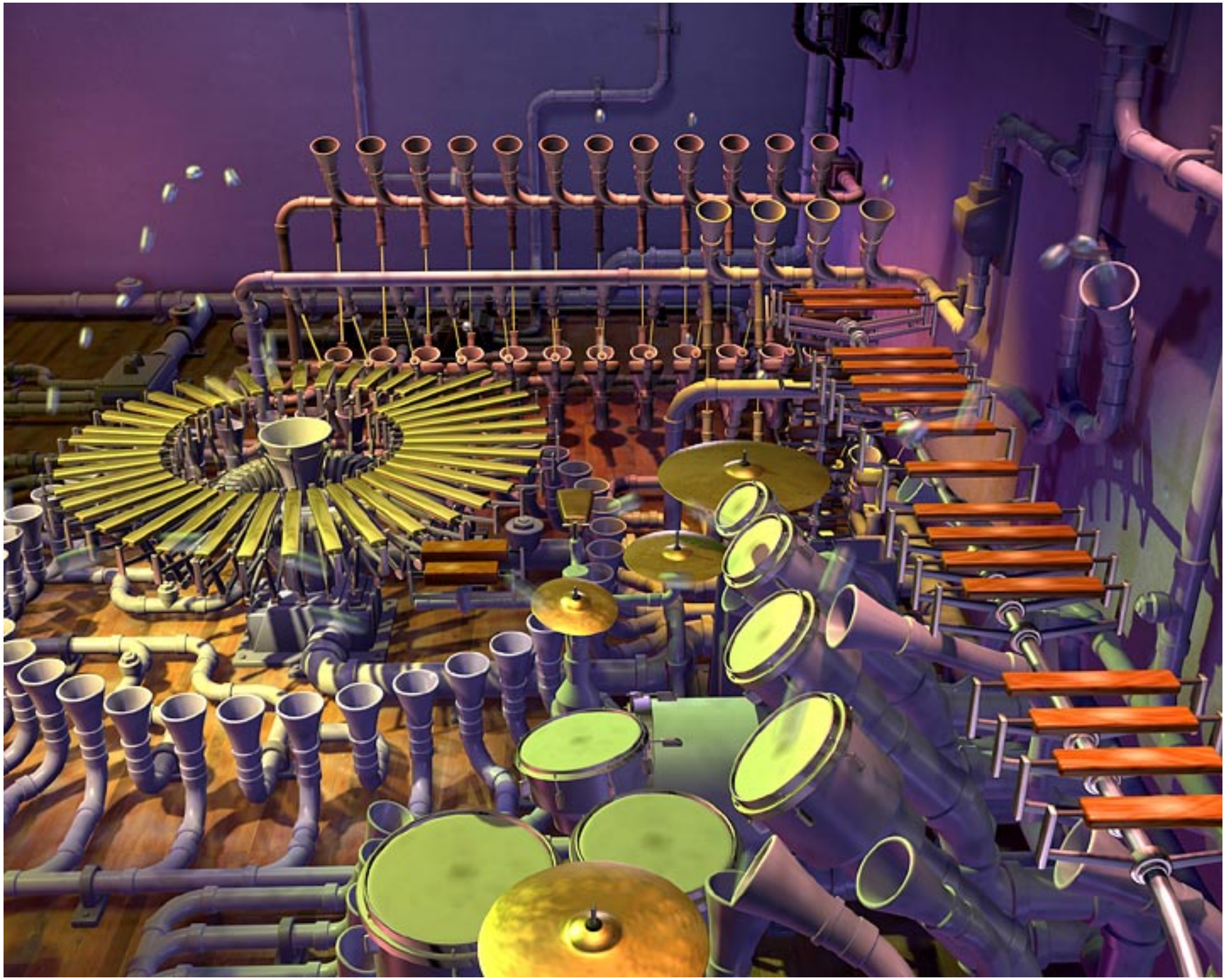
- Modeling
  - 2D and 3D representation
  - Curved surfaces
  - Procedural techniques
- **Rendering**
  - Realism
  - Speed
  - Non-realism
- Animation
  - Illusion of life
  - Motion capture
  - Keyframing
  - Physical simulation

Graphics Problems

- Modeling
  - 2D and 3D representation
  - Curved surfaces
  - Procedural techniques
- Rendering
  - Realism
  - Speed
  - Non-realism
- **Animation**
  - Illusion of life
  - Motion capture
  - Keyframing
  - Physical simulation



Pipe Dream: Procedural Animation



YOUR TO DO LIST

Course Survey

- **Due Thursday**

- Who are you?
- Why are you here?
- What do you want to learn?
- What is your background?

Assignment 1: Hello, 2D!

- 3 points in space
- Learn Processing (or other…) API
  - Drawing 2D shapes
  - Simple animation
  - Color manipulation
  - User input
- Prep for your 2D project