CS 428: Fall 2010

Introduction to Computer Graphics

Introduction and Overview
First things first...

- 3 (+1) slots left in CS 428 (as of 8/31/2010)
- Write me by 11:59pm today. subject: **CS428**
  - Your full name
  - Why you want to take this course in **max** three sentences
  - Grades for the CS 428 prerequisites
    - Calculus 2, Linear algebra, Java

- Keep it short!
- Will inform you on accept/reject by Friday
People

- **Instructor: Prof. Andrew Nealen**
  - CBIM (Bowser road, near student center)
    Room 21, nealen@cs.rutgers.edu
  - **Office hours:** Wednesday 5-7pm,
  - Best to contact me by email, or office hours

- **TA: Peter Borosan**
  - CBIM lab and Hill 250, 252 for office hours
  - Office hours: TBA (Peter will be back on 9/13)
  - pborosan@cs.rutgers.edu
Web

- Website (external access)
  - http://tinyurl.com/cs428-fall2010

- Everything else: Sakai
  - You should be able to see the “CS 428 Fall 2010” tab after logging in to https://sakai.rutgers.edu/portal with your RutgersID
  - Mailing list:
    cs428-fall2010@sakai.rutgers.edu
What’s required?

- Programming (Java, JOGL 1.0.0, JOGL 1.1.0)
- Math
  - linear algebra, some numerical computation
- Time commitment
  - This course is very work/code intense
  - You need to be aware of this!
  - How many other courses are you taking?
Textbook

*Computer Graphics with OpenGL, 3rd edition*
Donald Hearn and M. Pauline Baker,
Prentice Hall, 2004
Academic Integrity

- Read the web page!
  http://academicintegrity.rutgers.edu/integrity.shtml
- But basically:
  - You need to do your own thinking, writing, and programming
  - You *should* discuss the course material with other people in the class, but you cannot give away how to do the homework(s) or projects
Computer accounts

- We’ll mainly be using Hill 248, 250 and 252 (iLab a.k.a. the “cereal” named machines)
- Available on the course website
  - How to set up an account
  - How to get JOGL up and running
  - More...
- Access to the lab with your Rutgers ID
Programming projects
Grading

- Projects 60%
- Midterm 20%
- Final 20%

Late policy:
- tell me about problems in advance
- 50% credit for one day late (and no extra credit)
- 0% credit for 2+ days late
What this course is about

- Representations, computational models, and algorithms in computer graphics
- Using OpenGL on modern (programmable) graphics hardware
Representation + comp. models

- Shapes + materials + appearances
- Motions + behaviors
- **Representations:** specifications
- **Comp. models:** realizations
- Structure of problems
  - Spatial/temporal coherence. Sparsity.
- Optimization
  - Approximation. Pre-computation.

2D/3D models
images + video etc...

Algorithms:
simulation
direct solution

Andrew Nealen, Rutgers, 2010 9/1/2010
Topic overview

- Image formation and OpenGL
- Transformations and viewing
- Polygons and polygon meshes
  - Programmable pipelines
- Modeling and animation
- Rendering
Topic overview

- **Image formation and OpenGL**
  - Modeling the image formation process
  - OpenGL primitives, OpenGL state machine
- Transformations and viewing
- Polygons and polygon meshes
  - Programmable pipelines
- Modeling and animation
- Rendering
Topic overview

- Image formation and OpenGL
- Transformations and viewing
  - Linear algebra review, Homogeneous coordinates
  - Geometric + projective transformations
  - Viewing, Viewports, Clipping
- Polygons and polygon meshes
  - Programmable pipelines
- Modeling and animation
- Rendering
Topic overview

- Image formation and OpenGL
- Transformations and viewing
- **Polygons and polygon meshes**
  - 3D model/mesh representations
  - Piecewise linear shape approximations
  - Illumination and polygon shading
- Modeling and animation
- Rendering
Topic overview

- Image formation and OpenGL
- Transformations and viewing
- Polygons and polygon meshes
  - Programmable pipelines
- **Modeling and animation**
  - Procedural modeling and animation
- Rendering
Topic overview

- Image formation and OpenGL
- Transformations and viewing
- Polygons and polygon meshes
  - Programmable pipelines
- Modeling and animation

- **Rendering**
  - OpenGL rasterization: hidden surface removal, interpolation, texturing (some sampling theory)
  - Raytracing and radiosity
3D graphics programming
High-level view

Modeling

3D Objects

3D Models

approx. 25 Triangles

Rendering

Images

approx. 50 x 100 Pixels
3D graphics programming
High-level view

Modeling
3D Objects

Rendering
Images
- Interaction
- Animation

3D Models
- Scene
- Geometry
- Material
- Illumination
Making images in CG
Appearance
Shape
Motion
Movies
Tron (1982)
Movies
Luxo Jr. (1986)
Movies

The Matrix Revolutions (2003)
Movies
King Kong (2005)
Movies

Ratatouille (2007)
Movies

Wall-E (2008)
Video games
Team Fortress 2 (2007)
Video games

Resident Evil 5 (2009)
Video games
Spore (2008)
Computational Design
Scientific/Medical Visualization
Training and Education
Art