Week 1, Wed Jan 5

Introduction

Course Structure
- 40% programming projects
  - 10% project 1 (building beasties)
  - 10% project 2 (flying frenzy)
  - 20% project 3 (graphics game)
  - 25% final
- 20% midterms
  - 10% midterm 1 (week 6, Fri Feb 11)
  - 10% midterm 1 (week 11, Mon Mar 21)
- 15% written assignments
  - 5% each HW 1/2/3

Programming Projects
- structure
  - C++, Windows or Linux
  - OpenGL graphics library
  - GLUT for platform-independent windows/UI
  - face to face grading in lab
- project 1: building beasties
  - previous years: elephants, birds, poodles
  - Hall of Fame
- project 3: create your own game
  - Hall of Fame

Late Work
- 3 grace days
  - for unforeseen circumstances
  - strong recommendation: don’t use early in term
  - otherwise: 25% per 24 hours
- no work accepted after solutions handed out
- severe illness or crises, as per UBC rules
  - let me know ASAP (in person or email)
  - must also turn in form with documentation

Course Information
- course web page is main resource
  - http://www.ugrad.cs.ubc.ca/~cs314
  - updated often, reload frequently
- newsgroup is ubc.courses.cpsc.414
- note old course number still used
- readable on or off campus
- (no WebCT)

Labs
- normal labs
  - Mon 12:00-13:00, Tue 13:00-14:00, Tue 14:00-15:00, Wed 12:00-13:00, Thu 10:00-11:00, Fri 12:00-13:00
- special lab coverage hours
  - before project deadlines, will be posted
Teaching Staff
- instructor: Dr. Munzner
  - office hrs in CICSR 011
    - Wed 3:45-4:45
    - and extra lab hours as posted
- TAs: Dan Julius, Dana Sharon, Chen Yang
- contact email
  - {tmm,djulius,dsharon,cyang}@cs.ubc.ca
- use newsgroup not email for all questions that other students might care about

Plagiarism and Cheating
- don’t cheat, I will prosecute
- insult to your fellow students and to me
- programming and assignment writeups must be individual work
  - exception: project 3 can be team of two
- can discuss ideas, browse Web
- but cannot just copy code or answers
- cite all sources of information
  - web sites, study group members, books

Required Reading
- Fundamentals of Computer Graphics
  - Peter Shirley, AK Peters
- OpenGL Programming Guide, v 1.4
- OpenGL Architecture Review Board
  - v 1.1 available for free online
- readings posted on schedule page

Schedule (subject to change)
- wk 1:   Intro, Math Review
- wk 2:   OpenGL, Transformations
- wk 3:   Transformations, Viewing, Projections
- wk 4:   Lighting/Shading
- wk 5:   Shading, Rasterization
- wk 6:   Rasterization, Midterm Review, Midterm 1
- wk 7:   Hidden Surfaces, Clipping
- wk 8:   Textures
- wk 9:   Advanced Rendering, Picking
- wk 10:  Modelling, Midterm Review
- wk 11:  Midterm 2, TBD
- wk 12:  Visualization
- wk 13:  Animation, Final Review, CG in Movies

What is Computer Graphics?
- create or manipulate images with computer
- this course: algorithms for image generation

What is CG used for?
- graphical user interfaces
- modeling systems
- applications
- simulation & visualization
What is CG used for?
- movies
- animation
- special effects

What is CG used for?
- computer games

What is CG used for?
- images
- design
- advertising
- art

Real or CG?
http://www.alias.com/eng/etc/fakeorfoto/quiz.html

Real or CG?

Real or CG?
This Course

- We cover
  - Basic algorithms for
    - Rendering – displaying models
    - (Modeling – generating models)
    - (Animation – generating motion)
  - Programming in OpenGL, C++
- We do not cover
  - Art/design issues
  - Commercial software packages

Other Graphics Courses

- CPSC 424: Geometric Modeling
- CPSC 426: Computer Animation
- CPSC 514: Image-based Modeling and Rendering
- CPSC 526: Computer Animation
- CPSC 533A: Digital Geometry
- CPSC 533B: Animation Physics
- CPSC 533C: Information Visualization

Rendering

- Creating images from models
  - Geometric objects
    - Lines, polygons, curves, curved surfaces
  - Camera
    - Pinhole camera, lens systems, orthogonal
  - Shading
    - Light interacting with material
- Pixar Shutterbug series
  - Williams and Siegel using Renderman, 1990
  - www.siggraph.org/education/materials/HyperGraph/shutbug.htm

Modelling Transformation: Object Placement
Viewing Transformation: Camera Placement

Perspective Projection

Depth Cueing

Depth Clipping

Colored Wireframes

Hidden Line Removal
Complex Lighting and Shading

Texture Mapping

Displacement Mapping

Reflection Mapping

Modelling
- generating models
- lines, curves, polygons, smooth surfaces
- digital geometry

Animation
- generating motion
  - interpolating between frames, states
  - physical simulation (cloth)
  - motion capture (figure)