COMPUTER GRAPHICS
CS-314: Fall 2005
Instructor: Alla Sheffer

http://www.ugrad.cs.ubc.ca/~cs314
What is Computer Graphics?

- Generation of static/dynamic (realistic) images on computer
What is CG used for?

- GUI
  - Modeling systems
  - Applications
- Simulations & 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
What is CG used for?

- Virtual reality
What This Course Is About

- Basic **algorithms** for
  - Modeling – generating models
  - Rendering – displaying models
  - (Animation – generating motion)
- Programming in OpenGL, C++
What This Course is NOT About

- NOT covered:
  - Artistic and design issues
  - Usage of commercial software packages
Expectations

- Hard course!
  - heavy math
  - heavy programming
- Fun course!
  - graphics programming
    addictive, create great demos
- Programming prereq
  - CPSC 216 (Program Design and Data Structures)
  - Good knowledge of C++
- Math prereq
  - MATH 200 (Calculus III)
  - MATH 221/223 (Matrix Algebra/Linear Algebra)
Other graphics courses

- CPSC 424: Geometric Modeling
  - Not given this year
- CPSC 426: Computer Animation
  - Given in the fall
- CPSC 514: Image-based Modeling and Rendering
- CPSC 526: Computer Animation
- CPSC 533A: Digital Geometry
- CPSC 533B: Animation Physics
- CPSC 533C: Information Visualization
Policies
Teaching Staff

- Instructor: Alla Sheffer
  - Office hrs: ClCSR 011/X651, Tue 4-5pm
  - Contact info:
    - sheffa@cs.ubc.ca
    - do NOT use for
      - assignment related questions
      - anything else which might be relevant to other students

- Temporary instructor (Sep 13-23):
  - Dan Julius djulius@cs.ubc.ca

- TA: Stephen Ingram sfingram@cs.ubc.ca
Course Information

- **Up-to-date information:**
  - [http://www.ugrad.cs.ubc.ca/~cs314](http://www.ugrad.cs.ubc.ca/~cs314)
    - updated often, reload frequently
  - WebCT (follow link from course home page)
    - Bulletin board
    - REQUIRES INTERCHANGE ACCOUNT!
  - I assume that once information is posted on WebCT or web-page students know it within 2 workdays
Labs

- Will have **only 2 labs** (cutbacks + low student registration)
  - Option 1: Mon 13-14, Thu 15:30-16:30
  - Option 2: Mon 13-14, Wed 12-13
  - Option 3: Wed 12-13, Thu 15:30-16:30
- Vote
- Example problems in spirit of written assignments and exams + help with programming assignments
- Strongly recommend that you attend
Grading

- **Assignments (programming + theory):** 45%
  - OpenGL+Math 101 (3%)
  - 3D Transformations (13%)
  - Rendering pipeline (13%)
  - OpenGL Extravaganza (16%)

- **Two Quizzes:** (30%)
  - 15% each

- **Final Exam:** (25%)
Important Dates

- Assignment 0 due: Sep 23
- Assignment 1 due: Oct 14
- Assignment 2 due: Nov 4
- Assignment 3 due:
  - Theory: Nov 25
  - Programming: Nov 28

- Quiz 1: Oct 20
- Quiz 2: Nov 10
Course Organization

- Programming assignments:
  - C++, Windows or Linux
  - OpenGL graphics library / GLUT for user interface

- Face to face grading in lab
  - Opportunity to show all the “cool” extra stuff
  - Test that you do know what every piece of your code does

- Hall of fame – coolest projects from 2002 on
Late Work

- 3 grace days
  - for unforeseen circumstances
  - strong recommendation: don’t use early in term
  - handing in late uses up automatically unless you tell us

- Otherwise: 25% per 24 hours
  - no work accepted after solutions handed out

- Exception: severe illness or crisis, as per UBC rules
  - MUST
    - Get approval from me ASAP (in person or email)
    - Turn in form with documentation

http://www.ugrad.cs.ubc.ca/~cs314/Vsep2005/policies.html#illness
Regrading

- To request assignment or exam regrade:
  - Submit detailed written explanation - why you think the grader was incorrect for the particular problem that you are disputing
- I will regrade *entire* assignment:
  - thus even if I agree with your original request, your score may end up higher or lower
Literature (optional)

- **Fundamentals of Computer Graphics**
  - *Second edition*
  - Peter Shirley, A.K. Peters

- **OpenGL Programming Guide**
  - J. Neider, T. Davis and W. Mason, Addison-Wesley
Learning OpenGL

- This is a graphics course using OpenGL
  - not a course ON OpenGL
- Upper-level class: learning APIs mostly on your own
  - only minimal lecture coverage
    - basics, some of the tricky bits
  - OpenGL Red Book
- many tutorial sites on the web
  - nehe.gamedev.net
Plagiarism and Cheating

- Don’t cheat, I will prosecute
  - insult to your fellow students and to me
- Theoretical assignments are individual work
- Programming assignments (when specified) can be done in pairs
- Can discuss ideas, browse Web
- But cannot copy code or answers
- **Must** be able to explain algorithms during face-to-face demo
  - or no credit for that assignment, possible prosecution
Citation

- Cite all sources of information in assignment’s README
  - web sites, study group members, books

http://www.ugrad.cs.ubc.ca/~cs314/Vsep2005/policies.html#plag
Rendering in 2D

- Raster display - discrete grid of elements
- Terminology
  - **Pixel**: basic element on device
  - **Resolution**: number of rows & columns in device
    - Measured in
      - Absolute values (1K x 1K)
      - Density values (300 dots per inch)
- **Screen Space**: Discrete 2D Cartesian coordinate system of the screen pixels
Basic Rendering – 2D

- Algorithms for:
  - Scan Conversion
    - Draw (lines)
      - Anti-Aliasing
    - Fill (polygons)
    - Clipping
  - Color
3D Graphics Components

- Geometric Modeling
  - polygons,
  - smooth surfaces,
  - etc...

- Rendering
Modeling Transformation: Object Placement
Viewing Transformation: Camera Placement
Perspective Projection
Hidden Line Removal
Hidden Surface Removal
Per-Polygon Shading
Specular Reflection
Phong Shading
Complex Lighting and Shading
Texture Mapping
Shadows (+Displacement Mapping)
Reflection Mapping
Animation

- Generating motion
  - Frame/State interpolation
  - Physical simulation
  - Motion capture
Lecture Syllabus

- Introduction + Math Review + OpenGL + Rendering Pipeline (week 1)
- Transformations (week 2/3)
- Rasterization (week 4/5)
- Clipping (week 5)
- Hidden Surface Removal (week 6)
- Review & Quiz (week 7)
  - Quiz: Oct 20
- Lighting Models (week 8)
- Texture mapping (week 9)
- Review & Quiz (week 10)
  - Quiz: Nov 10
- Shadows (week 11)
- Ray Tracing (week 11/12)
- Geometric Modeling (week 12/13)
- Review (last lecture)