Instructor

• Willem A (Vlakkies) Schreüder

• Email: willem@prinmath.com
  - Begin subject with 4229 or 5229
  - Resend email not answered promptly

• Office Hours:
  - Before and after Class
  - By appointment

• Weekday Contact Hours: 6:30am - 9:00pm
Course Objectives

• Class: Theory and principles
  – Attendance is encouraged
• Assignments: Practical OpenGL
  – Applications
• No tests or exams

• By the end of the course you will:
  – Be conversant in computer graphics principles
  – Be well versed in the use of OpenGL
  – Understand what OpenGL does internally
Course Outline

• Basics (1/3)
  − Projections, transformations, clipping, rendering, text, color, hidden edge and surface removal, and interaction

• Advanced (1/3)
  − Illumination, shading, transparency, texture mapping, parametric surfaces, shaders

• Project (1/3)
  − Whatever you're interested in: games, modeling, visualization, 'Google Earth', ....
Why OpenGL?

- Modern, widely used and actively supported
  - Games
  - 3D visualization
- Cross platform
  - Windows
  - Mac
  - *NIX
- Open source and vendor implementations
  - MESA 3D (source code available)
- Many language bindings
Assumptions

• You need to be fluent in C
  – Examples are in C
  – You can do assignments in any language
    • I may need help getting it to work on my system

• You need to be comfortable with linear algebra
  – Matrix and Vector multiplication
  – Dot and cross products
  – Rotation matrices
Grading

• Satisfactory complete all assignments => A
  – The goal is to impress your friends
• Assignments must be submitted on time unless prior arrangements are made
  – Due Thursday evening 11:59 pm
  – Grace period until Friday morning at 06:30am
• Assignments must be completed individually
  – Stealing ideas are permitted
  – OpenGL code fragments from the web may be used
• OpenGL: A Primer, 3/E
  - Edward Angel
  - An excellent and very accessible introduction to OpenGL - and inexpensive
  - Third edition adds new material including shaders
  - Recommended but not required

• Computer Graphics: Principles & Practice (2ed)
  - Foley, van Dam, Feiner & Hughes
  - Avoid 1ed (Pascal), 2ed also a bit dated
  - Get it if you want to know more of the theory
Other Texts

• OpenGL Programming Guide (5ed)
  − Shreiner, Woo, Neider & Davis
  − “OpenGL Red Book”
  − Download previous editions as PDF

• OpenGL SuperBible: Comprehensive Tutorial and Reference (4ed)
  − Wright, Lipchak & Haemel
  − Good all-round theory and applications
And More Texts

- **OpenGL Shading Language (2ed)**
  - Randi J. Rost
  - “OpenGL Orange Book”
  - Introduces both OpenGL and Shaders

- **OpenGL Reference Manual (4ed)**
  - OpenGL Architecture Review Board & Dave Shreiner
  - “OpenGL Blue Book”
  - Official Reference Document to OpenGL, Version 1.4
  - A bit dated, very similar to man pages
OpenGL Resources

• www.google.com
  – Need I say more?

• www.opengl.org
  – Code and tutorials

• nehe.gamedev.net
  – Excellent tutorials

• www.mesa3d.org
  – Code of “internals”
Assignment 0

• Due: **Wednesday** Sep 5, 2007

• Sign up with moodle.cs.colorado.edu
  - Enrollment key: 42295229
  - A picture will help me learn your names

• Submit
  - Your name and study area
  - Platform (Hardware, Graphics, OS, ...)
  - Background and interests in computer graphics
  - Project ideas (if you have one already)
My information

• Mathematical modeling and data analysis
  – PhD Computational Fluid Dynamics [1986]
  – PhD Parallel Systems (CU Boulder) [2005]
  – President of Principia Mathematica

• Use graphics for scientific visualization

• Open source bigot

• Program in C, C++, Fortran and Perl
Assignment 1

• Due: Thursday Sep 13, 2007
• Get OpenGL to work on your platform
  – Compile and run `gears.c`
  – Report frame rate for 1x1, 300x300 and full screen
  – Explain your results
• If you are on an X based (*NIX) platform:
  – Run glxinfo and check if `direct rendering: yes`
  – Look into enabling hardware support
Assignment 2

• Due: Sep 20, 2007

• Write an OpenGL based visualization of the Lorenz Attractor
  − At a minimum show a static line path in 3D
  − Add rotation using cursor keys
  − Use your imagination

• The purpose is scientific visualization
  − Do some science

http://mathworld.wolfram.com/LorenzAttractor.html
Nuts and Bolts

• Complete assignments on any platform
  − Assignments reviewed under Fedora Core
  − Set #ifdef so I can compile and run it

• Submit using moodle.cs.colorado.edu
  − ZIP or TAR
  − Name executables hw1, hw2, ...
  − Set makefile so I can do make LINUX=1
  − Set window title to Assignment X: Your Name

• Include number of hours spent on assignment
A few hints

• My machine runs Fedora Core x86_64
  – gcc/g++ with Mesa3D & GLX
    • -Wall is a really good idea
  – case sensitive file names
  – int=32bit, long=64bit
  – little-endian
  – fairly good performance

• How to make my life easier
  – Try it in CSEL or a Linux box
  – Stick to C/C++ unless you have a good reason
The Importance of Graphics:
100 Values between 0 and 1
100 Values between 0 and 1
The Importance of Graphics
Graphic Design

• 2D vs. 3D
  − Cool vs. informative

• Edward R. Tufte
  − Visual Explanations
  − Envisioning Information
  − The Visual Display of Quantitative Information
  − Beautiful Evidence
Saturn from Cassini Probe
Colorado Fall Colors
What is wrong with this picture?
In the beginning....
Storage Tube Terminals
Storage Display Images
Color: Multiple Pen Plotters
Raster Graphic Terminals
Color Inkjets
Workstations: Apollo DN 330
12 MHz 68020, 3MB RAM, 70MB disk
Plotting Packages

- PLOT-10: Tektronix 4010 graphics
- PLOT88: PC graphics
- DISSPLA: NCAR graphics
- GINO: Portable graphics
- DIGLIB: LLNL device-independent, open source
- GKS: Graphics Kernel System (2D vector)
- PHIGS: 3D Interactive Graphics
The rise of OpenGL

• Originated as SGI IrisGL
• Vendor-neutral OpenGL controlled by ARB
• Hides the details of hardware
  – Software emulation when necessary
  – Hardware acceleration when possible
• Supports 2D to advanced 3D graphics
• Portable to most hardware and OS with WGL, AGL and GLX
Gaming and Graphics

- Text based/ASCII graphics (Pong, PacMan)
- 2D monochrome line graphics (Astroids)
- 2D images & sprites (Mario)
- 3D graphics
  - Flight Simulators (2D -> 3D)
  - First Person Shooters
  - Multi-player games
- Games push the envelope
  - Realism
  - Speed