CSCI 4229/5229 Computer Graphics Summer 2008

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 highly 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 Saturday evening 11:59 pm
  - Grace period until Sunday morning at 08:00am
- Assignments must be completed individually
  - Stealing ideas are permitted
  - OpenGL code fragments from the web may be used

Text

- 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 & Huges
  - 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 “in ternals”

Assignment 0

- Due: **Wednesday** Jun 4, 2008 at **noon**
- 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

Things to do before the next class

- Get OpenGL to work on your platform
  - Compile and run *Hello World* examples
- If you are using Windows
  - Link in my *glWindowPos* code
- If you are on an X based (*NIX) platform:
  - yum install freeglut-devel
  - Run glxinfo and check if *direct rendering: yes*
- OS/X based on OpenGL
  - Fink provides free development tools
- **You need to get this working by Thursday**

Assignment 1

- Due: Saturday June 7, 2008 at 23:59
- 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
- Example 6 is your friend

Assignment 2

- Due: Saturday June 14, 2008 at 23:59
- Write an program to visualize a 3D scene
- Scene must consist of solid 3D objects
  - You must create some solid 3D objects yourself
  - You must replicate some generic objects
- Scene must be viewable from different vantage points under user control
- Scene must be viewable using both orthogonal and perspective projections

Assignment 3

- Due: Saturday June 21, 2008 at 23:59
- Write an program to visualize a 3D scene with lighting and textures
  - Add lighting and textures to Assignment 2
  - Think about lighting and textures when you create the scene in Assignment 2

Project

- Should be a program with a significant graphics component
  - Something useful in your research/work?
  - Graphical front end to simulation
  - Graphical portion of a game
- Deadlines
  - Proposal: Monday June 23 (earlier is better)
  - Review: Saturday June 28 (progress report)
  - Final: Thursday July 3 (earlier for show and tell)

Nuts and Bolts

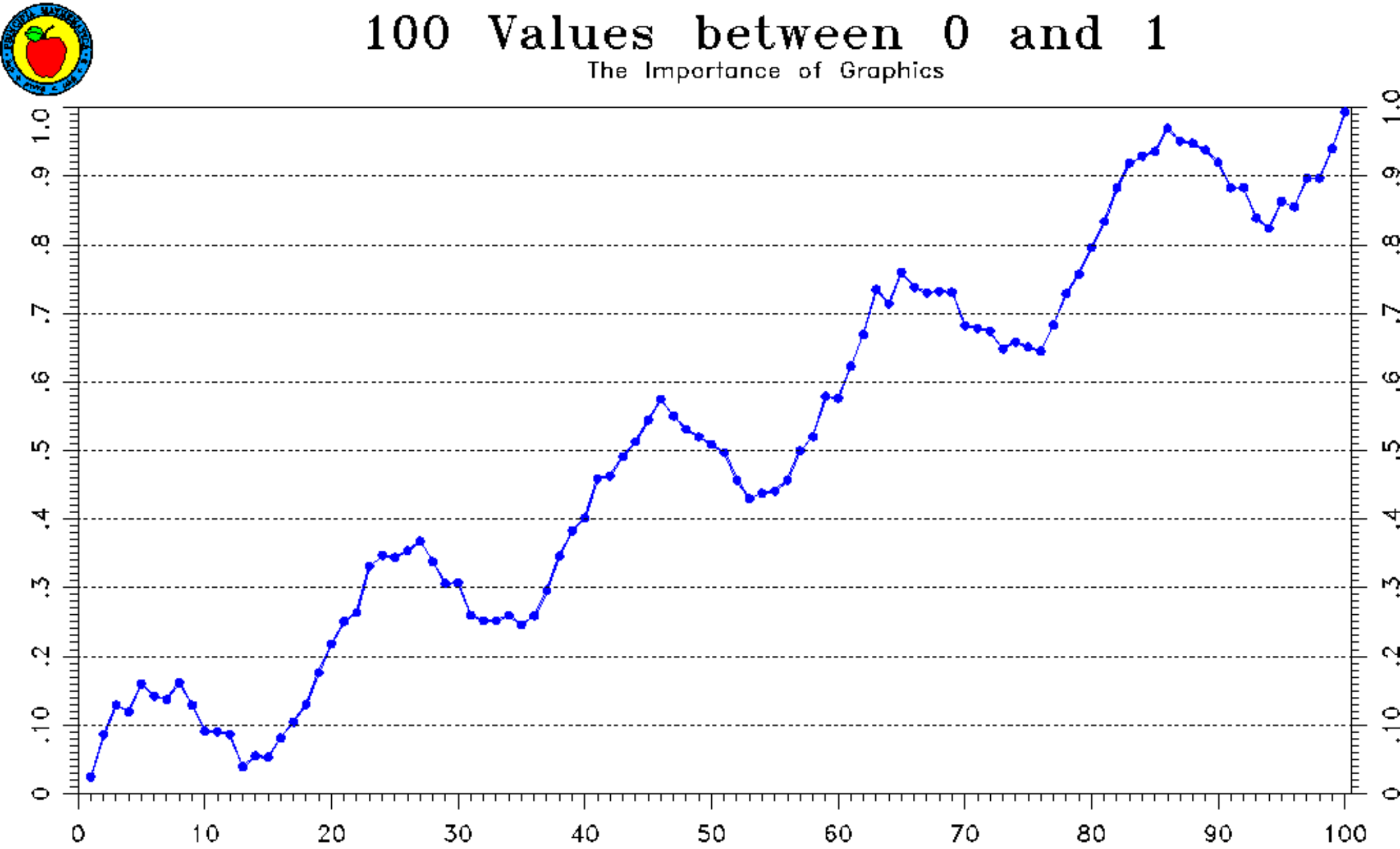
- Complete assignments on any platform
  - Assignments reviewed under CentOS 5.1
  - Set #ifdef so I can compile and run it
- Submit using moodle.cs.colorado.edu
  - ZIP or TAR
  - Name executables hw1, hw2, ...
  - Set makefile so I can do *make LINUX=1*
  - Set window title to *Assignment X: Your Name*
- Include number of hours spent on assignment

A few hints

- My machine runs Linux 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
- **Maintain thy backups…**

The Importance of Graphics: 100 Values between 0 and 1

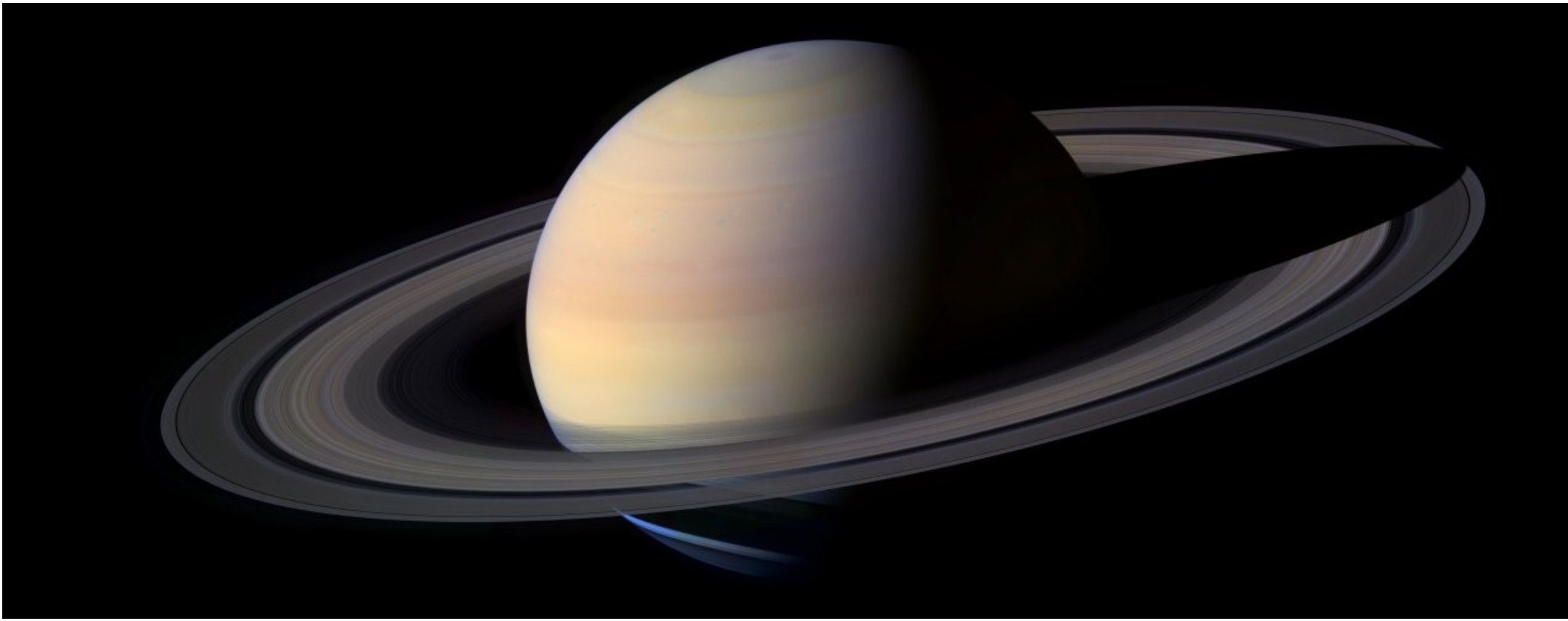
0.024	0.086	0.129	0.119	0.160	0.142	0.137	0.162	0.129	0.091
0.090	0.086	0.039	0.055	0.053	0.081	0.104	0.130	0.176	0.218
0.251	0.264	0.331	0.347	0.344	0.354	0.368	0.338	0.306	0.307
0.260	0.252	0.252	0.260	0.246	0.259	0.296	0.346	0.383	0.402
0.459	0.463	0.491	0.513	0.544	0.575	0.550	0.531	0.520	0.509
0.497	0.457	0.430	0.438	0.441	0.457	0.500	0.520	0.579	0.576
0.623	0.669	0.735	0.714	0.760	0.738	0.730	0.732	0.731	0.682
0.678	0.674	0.648	0.658	0.651	0.645	0.683	0.729	0.757	0.796
0.834	0.883	0.919	0.929	0.936	0.970	0.951	0.948	0.938	0.920
0.883	0.883	0.839	0.824	0.863	0.855	0.897	0.897	0.940	0.994


100 Values between 0 and 1
The Importance of Graphics
0
.10
.2
.3
.4
.5
.6
.7
.8
.9
1.0
0
10
20
30
40
50
60
70
80
90
100

Graphic Design

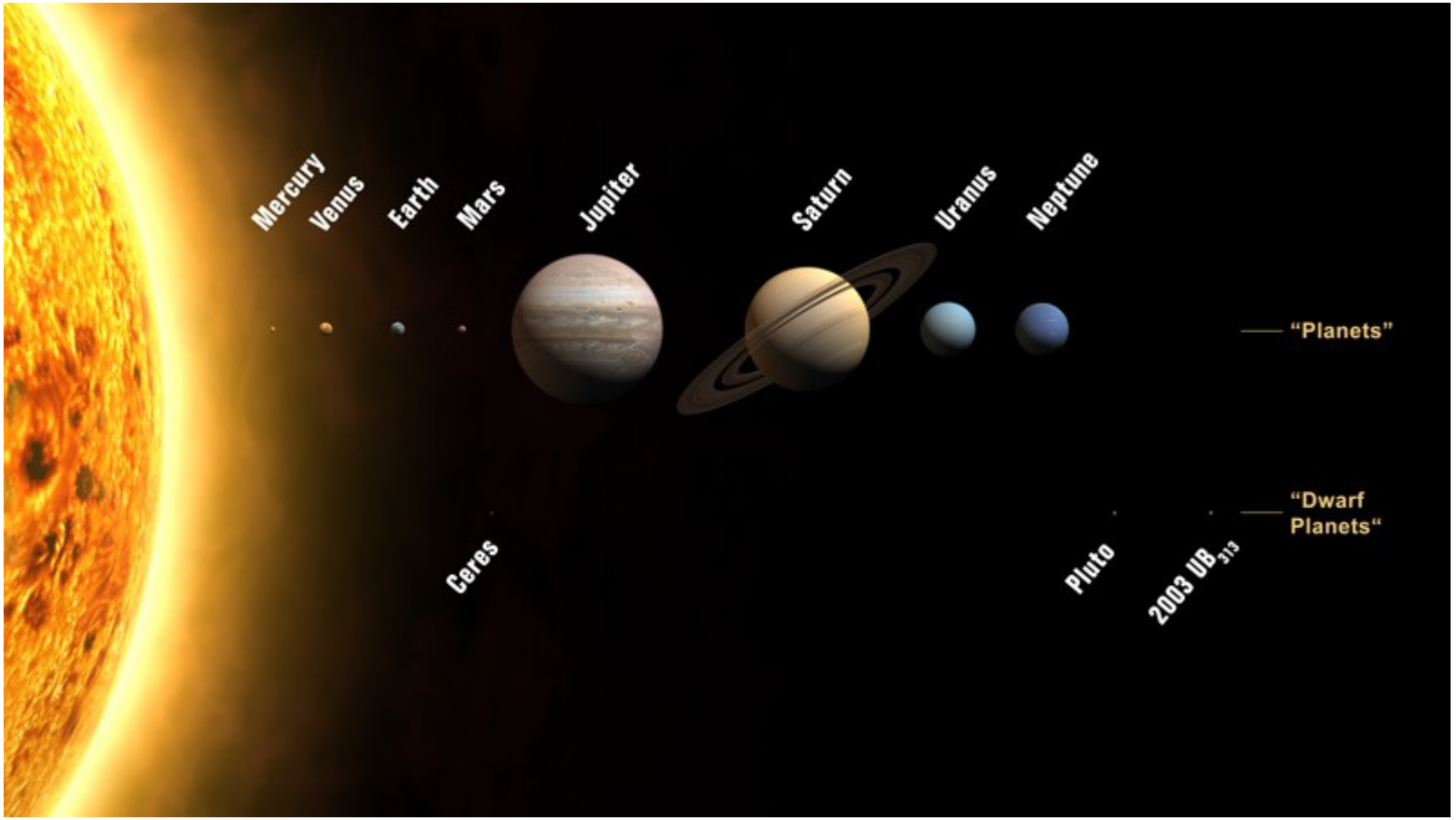
- 2D vs. 3D
  - Cool vs. informative
- Edward R. Tufte
  - Visual Explanations
  - Envisioning Information
  - The Visual Display of Quantitative Information
  - Beautiful Evidence

Saturn from Cassini Probe



Colorado Fall Colors

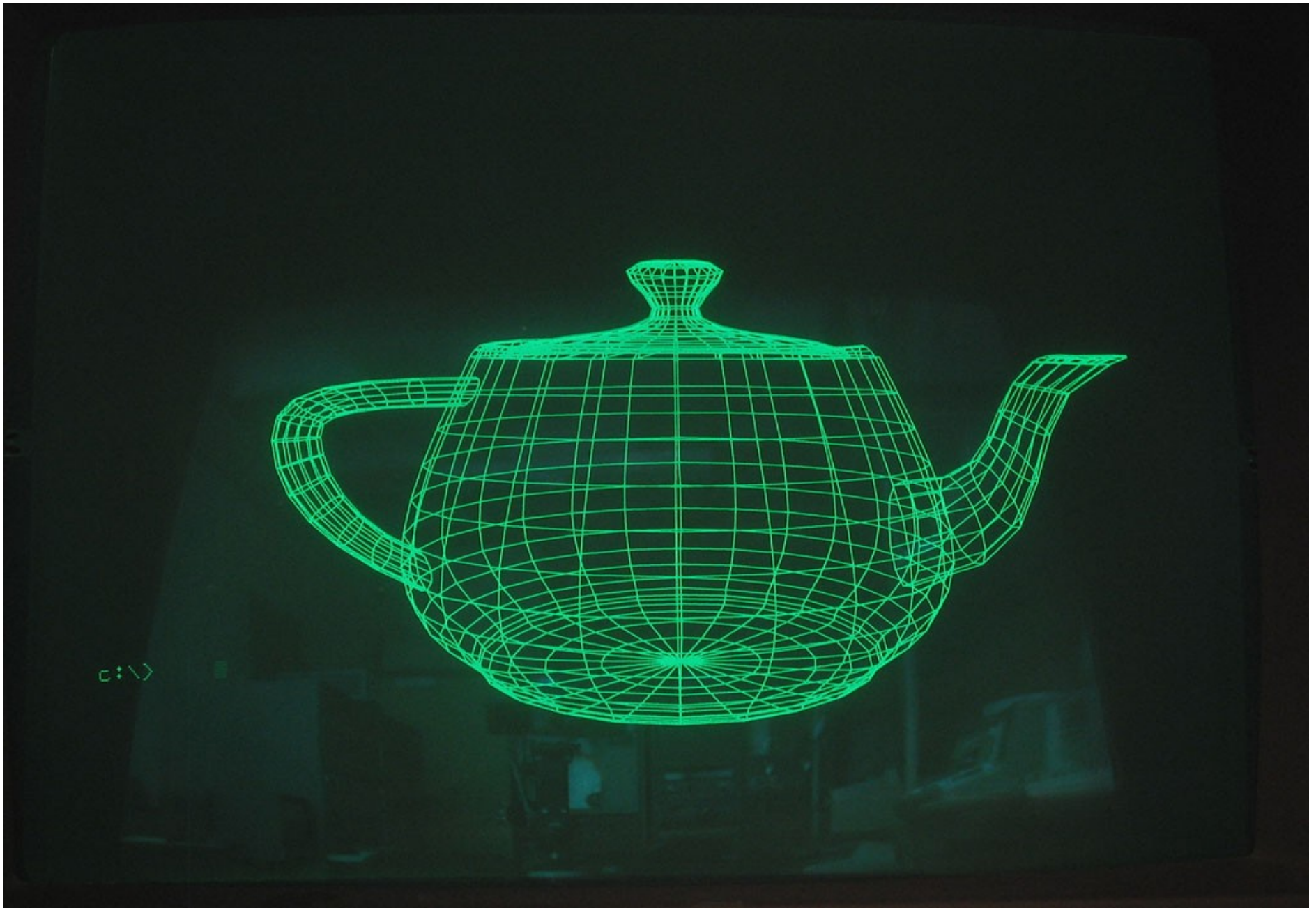
What is wrong with this picture?



In the beginning....

Storage Tube Terminals

Storage Display Images

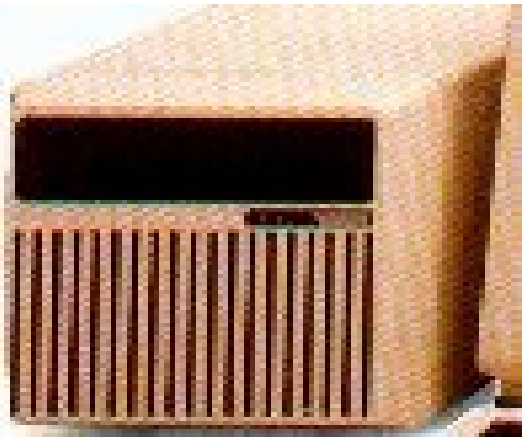


Color: Multiple Pen Plotters

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