

**CSCI 4229/5229**  
**Computer Graphics**  
**Fall 2025**

# 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

# Class Attendance

- Class Attendance in person is expected
  - Amplifies expectations and issues related to assignments
  - Covers additional examples not on Canvas
- If you feel ill, attend via Zoom
  - Email me by noon if - I will only do Zoom when requested

# 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
  - iPhone, Android and WebGL
- Open source and vendor implementations
  - MESA 3D (source code available)
- Many language bindings

# Instructor

- Willem A (Vlakkies) Schreüder
- Email: vlakkies@colorado.edu
  - Begin subject with 4229 or 5229
    - I have a draconian mail filter
  - Resend email not answered promptly
- Office Hours:
  - Monday 3-4pm via Zoom
  - Thursday 2:00-3:00pm in person or by Zoom
  - Email me to set up Zoom at other times
- Weekday Contact Hours: 6:30am - 9:00pm

# Assumptions

- You need to be fluent in C
  - Examples are in C
  - You need to know how to program and compile
  - You can do assignments in any language
    - I may need help getting it to work on my system
    - Use C or C++ unless you have a good reason
- You need to be comfortable with linear algebra
  - Vectors, surfaces, normals
  - 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
  - Most due Wednesday evening 11:59 pm
    - Extensions must be requested before due date
    - Grace period until Thursday morning at 08:00am
    - **Emailed assignments or attached as comments on Canvas will not be accepted**
- Assignments must be completed individually
  - Stealing ideas are permitted
  - OpenGL code from the web may be used
  - Make it your own and improve on it



# Grading Expectations

- Code reuse is acceptable
  - Give credit where it is due
  - You take responsibility for errors in reused code
  - You need to make a substantial improvement
    - I'm looking to see that you have insight in the material and put in a significant effort
    - Simply turning in downloaded code with minor changes is unacceptable and an Honor Code violation
- No grade => respond to my comments and resubmit
- **Grade <100 means NOT SATISFACTORY (not an A)**

# Example Programs

- Illustrates specific aspects
  - mode variable heavily overloaded
  - one concept at a time
- Designed to be a starting point
  - you are expected to improve on it
  - cut and paste will not get you an A
- **The course is not intended to teach OpenGL, but rather the principles underlying graphics**

# Text

- OpenGL Programming Guide (9ed)
  - Shreiner et al.
  - “OpenGL Vermillion Book”
  - Older edition was the “OpenGL Red Book”
  - Download previous editions as PDF
  - Recommended but not required

# Other Texts

- OpenGL: A Primer, 3/E
  - Edward Angel
  - Excellent and very accessible
  - Inexpensive
  - Third edition adds new material (shaders)
- OpenGL SuperBible: Comprehensive Tutorial and Reference (7ed)
  - Sellers, Wright & Haemel
  - Good all-round theory and applications
  - 6e & 7e is all OpenGL 4 which is a challenge

# Theoretical text

- Computer Graphics: Principles & Practice (3ed)
  - Hughes, van Dam, McGuire, Sklar, Foley, Feiner & Akeley
  - Avoid 1ed (Pascal), 2ed (very dated)
  - Get it if you want to know more of the theory

# Embedded OpenGL texts

- OpenGL ES 3.0 Programming Guide
  - Ginsburg & Purnomo
  - OpenGL Embedded Systems (iPhone & Android)
  - Subset of OpenGL, 1.3 and 2.0 very different
  - ***Not recommended for beginners***
- iPhone 3D Programming
  - Philip Rideout (O'Reilly series)
  - iPhone specific, but C/C++ oriented so translates well to Android (using the NDK)
  - My personal favorite for portable OpenGL ES

# OpenGL Resources

- Safari
- [www.google.com](http://www.google.com)
  - Need I say more?
- [www.opengl.org](http://www.opengl.org)
  - Code and tutorials
- [nehe.gamedev.net](http://nehe.gamedev.net)
  - Excellent tutorials
- [www.mesa3d.org](http://www.mesa3d.org)
  - Code of “internals”
- Class forum

# Assignment 0

- Due: Wednesday August 27 at 11:59pm
- Find the course on Canvas
- Submit
  - Your name and study area
  - Platform (Hardware, Graphics, OS, ...)
  - Background and interests in computer graphics
  - Project ideas (if you have one already)
  - Distance students let me know about special circumstances and schedules
  - Do office hours work for you



# My information

- Mathematical modeling and data analysis
  - PhD Computational Fluid Dynamics [1986]
  - PhD Parallel Systems (*CU Boulder*) [2005]
- Use graphics for scientific visualization
- Open source bigot
- Program in C, C++, Fortran and Perl

# Assignments

Week 1: Who Am I

Week 2: Compile and run Gears

Week 3: Visualizing the Lorenz Attractor

Week 4: Drawing Scene in 3D

Week 5: Orthogonal/Perspective Projection

Week 6: Lighting

Week 7: Lighting and Textures

Week 8: Project Proposal

Week 10: Project progress report

Week 12: Project Review

Week 14: Project Final

*Complete assignments early if you want*

# How to get started

- Get OpenGL to work on your platform
  - *Installing OpenGL* on Canvas
  - Compile and run *Hello World* examples
- If you are using Windows
  - Install MSYS2 and use pacman
  - Compile with -DUSEGLEW (see my examples)
- If you are on an X based (\*NIX) platform:
  - yum install freeglut-devel
  - apt-get install freeglut3-dev
  - Run glxinfo and check if *direct rendering: yes*
- OS/X based on OpenGL
  - Xcode command line
  - Use homebrew to install glfw, SDL, etc

# Assignment 1

- Due: Wednesday September 3 at 23:59
- Compile and run *gears* and answer a few questions
- This ensures that you have a working OpenGL environment on your machine
- Practice creating a *makefile*
  - After this I expect correct makefiles

# Assignment 2

- Due: Wednesday September 10 at 23:59
- Write an OpenGL based visualization of the Lorenz Attractor
  - At a minimum show a static line path in 3D
  - User control of attractor parameters
  - 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 3

- Due: Wednesday September 17 at 23:59
- Write an program to visualize a 3D scene
- Scene must consist of solid 3D objects
  - You must create objects yourself
    - no GLU/GLUT or imported objects
  - You must replicate some generic objects
- Scene must be viewable from different vantage points under user control
- *Add 3D objects to Assignment 2*

# Assignment 4

- Due: Wednesday September 24 at 23:59
- Write an program to visualize a 3D scene using both orthogonal and perspective projections as well as first person navigation
- *Add perspective projection to Assignment 3*
- *I don't provide a first person example*

# Assignment 5

- Due: Wednesday October 1 at 23:59
- Write an program to visualize a 3D scene with lighting
  - Make the light move to show lighting effects
  - Select solid objects that show lighting effects
  - Still no glu/glut or imported objects
- *Add lighting to Assignment 4*
- ***WARNING: THIS IS A VERY DIFFICULT ASSIGNMENT***



# Assignment 6

- Due: Wednesday October 16 at 23:59
- Write an program to visualize a 3D scene with lighting and textures
- End of progression of homework assignments
- *Add textures to Assignment 5*

# 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
  - Expect more from graduate students
- Deadlines
  - Proposal: Wednesday Oct 22
  - Progress: Wednesday Nov 5 (progress report)
  - Review: Wednesday Nov 19 (must run)
  - Final: Friday Dec 5 (no resubmit)

# Project Grading

- Half the total grade for the class
- The grade assigned for the ***final*** submission is what is counted
- Grades assigned for the progress report and the review are my assessment of what that final grade will be, and is not counted towards the class grade
  - Canvas is not smart enough to do this, so don't go by the totals it gives you

# Topics for CSCI 4239/5239

## Advanced Computer Graphics

- Shaders
  - Explicitly programming the GPU
- Embedded Systems
  - iPhone, Android, WebGL
- GPU work threads (CUDA & OpenCL)
- Ray Tracing

# Nuts and Bolts

- Complete assignments on any platform
  - Assignments reviewed under Ubuntu 22.04 LTS
  - Set `#ifdef` so I can compile and run it
- Submit using Canvas
  - Single ZIP file
  - Name executables hw1, hw2, ...
  - Create a makefile so I can do *make clean;make*
  - Set window title to *Assignment X: Your Name*
- Include number of hours spent on assignment
- *Check my feedback and resubmit if requested*
- This is a BIG class, **PLEASE** submit cleanly

# A few hints

- My machine runs Linux x86\_64
  - gcc/g++ with nVidia & GLX
    - -Wall is a **really** good idea
  - case sensitive file names
  - int=32 bit, long=64 bit
  - little-endian
  - fairly good performance
- How to make my life easier
  - Try it on a Linux box
  - Stick to C/C++ unless you have a good reason to use something else
- ***Maintain thy backups...***

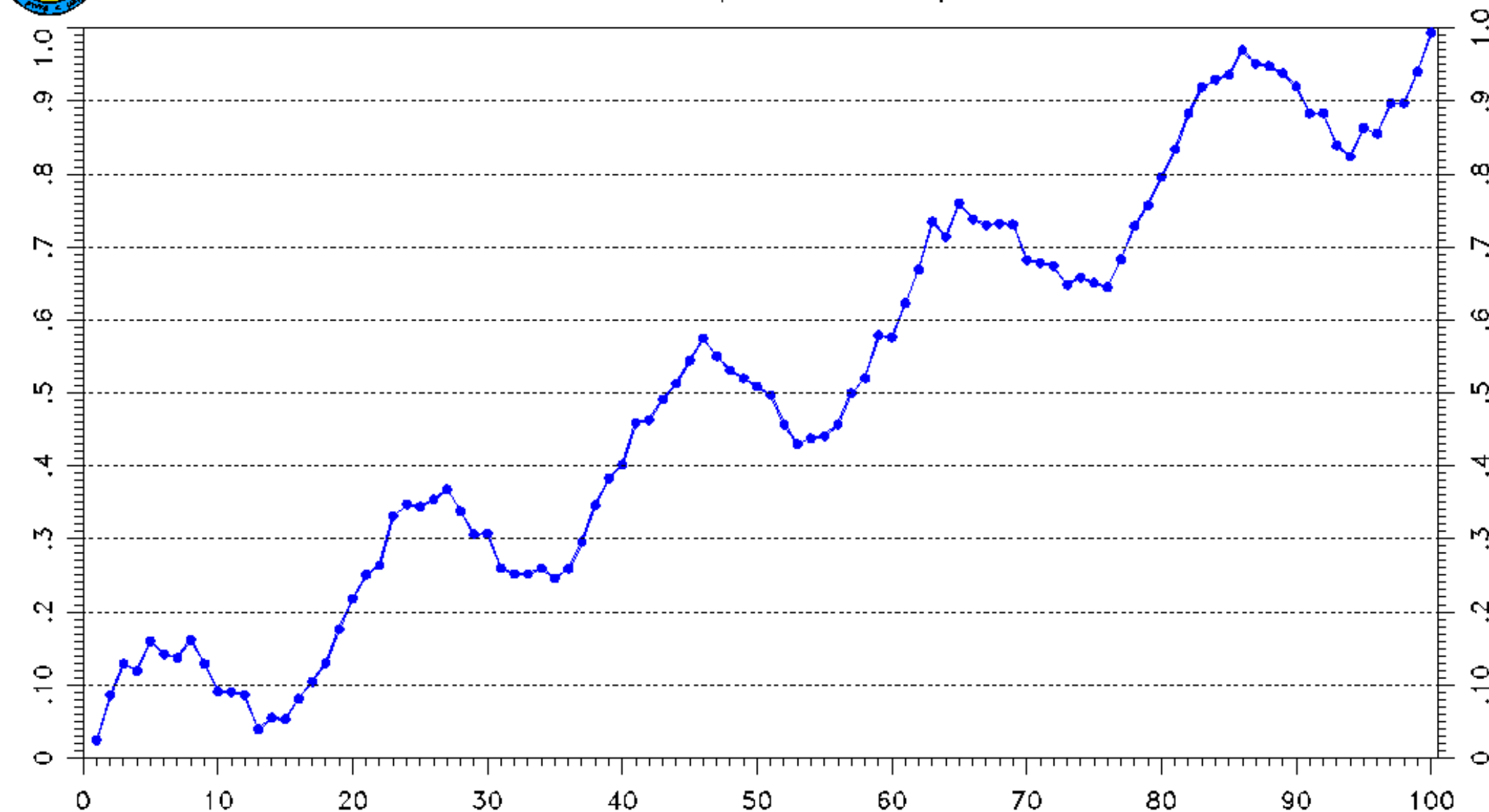
# The Importance of Graphics: 100 Values between 0 and 1

|       |       |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 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





# 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



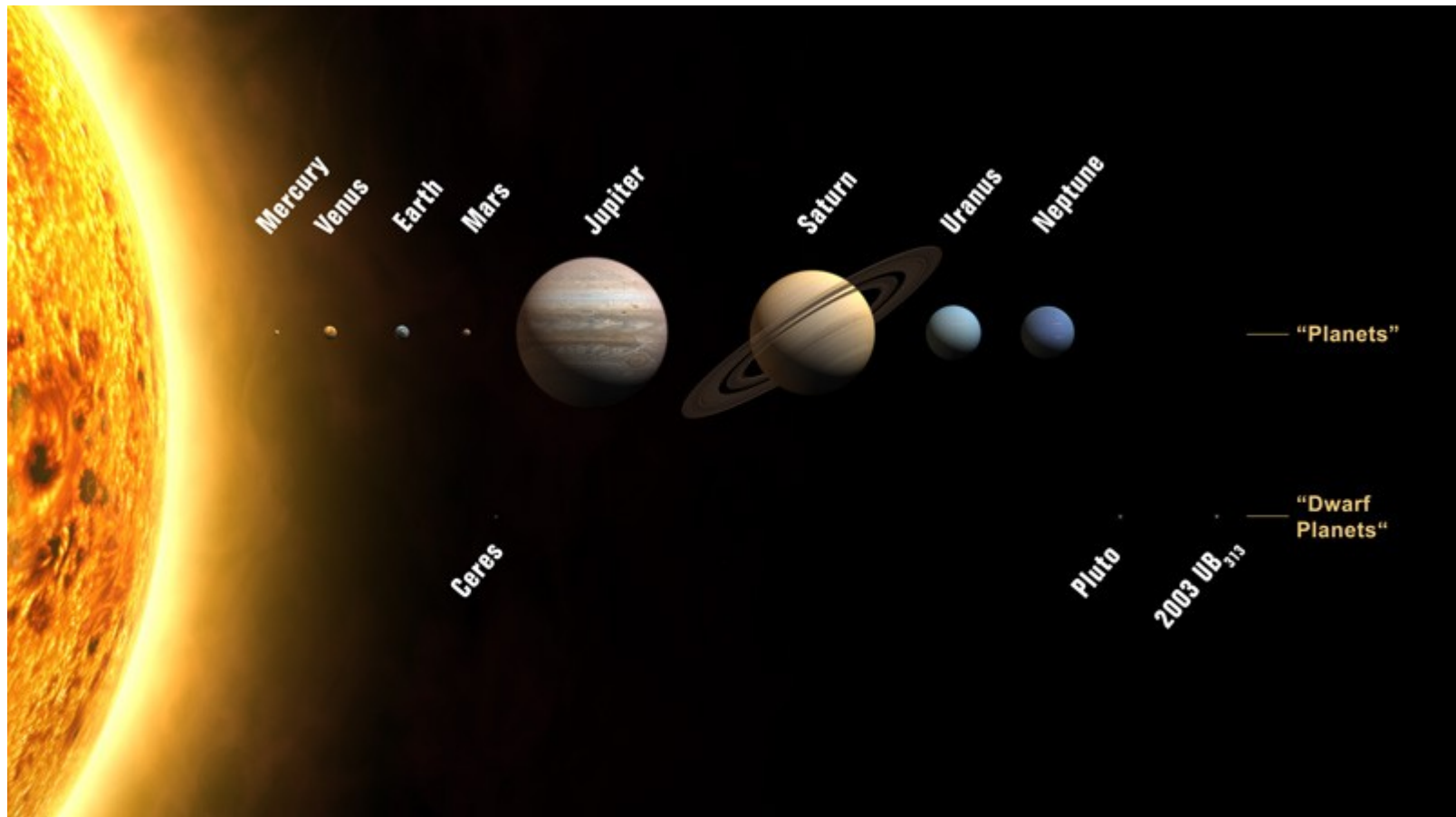


# Colorado Fall Colors





# What is wrong with this picture?



# In the beginning....





# Storage Tube Terminals



# Storage Display Images



# Color: Multiple Pen Plotters





# Raster Graphic Terminals

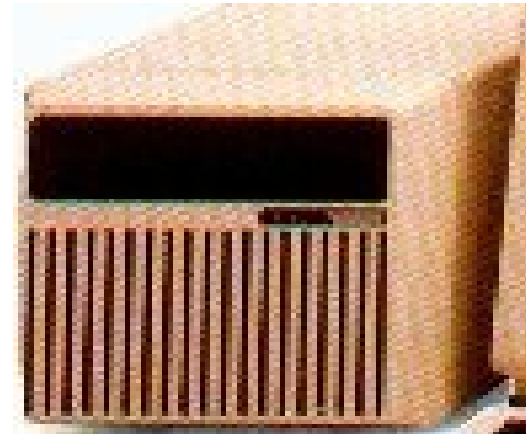


# Color Inkjets



# Workstations: Apollo DN 330

12 MHz 68020, 3MB RAM, 70MB disk



# Workstation, Desktop, Laptop, Phone, Communicator..

