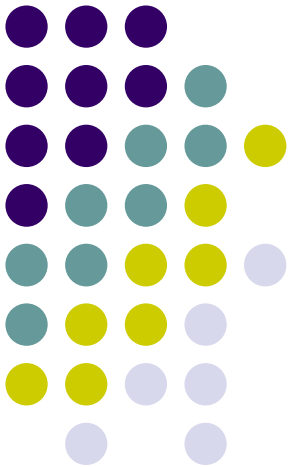
Computer Graphics
CS 543 – Lecture 1 (Part I)

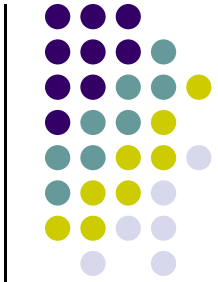
Prof Emmanuel Agu

Computer Science Dept.

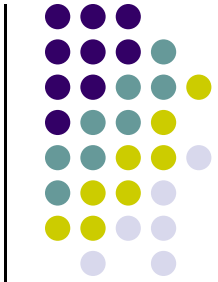
Worcester Polytechnic Institute (WPI)



About This Course



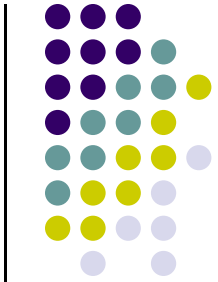
- Computer graphics: algorithms, mathematics, data structures ..... that **computer uses to generate PRETTY PICTURES**
- Techniques evolved over years, standardized into programmable libraries
- **OpenGL:** graphics library, open source!
- Future job: just program OpenGL (programmer's view)
- CS program: wants you to learn concepts, OpenGL Internals
- **We shall combine:**
  - **Programmer's view:** Learn how to program parts OpenGL
  - **Under the hood:** Learn how OpenGL is implemented, underlying algorithms, math, data structures (study OpenGL parts as concrete example)



About This Course

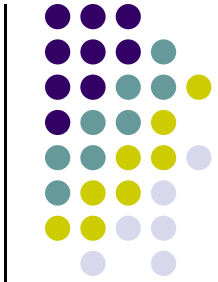
- Course about Computer Graphics
- Course is NOT
  - just about programming OpenGL
  - a comprehensive course in OpenGL. (Only select OpenGL parts will be covered)
  - about using packages like Maya, Photoshop
- Class is concerned with:
  - How to build graphics tools
  - Underlying mathematics
  - Underlying data structures
  - Underlying algorithms
- This course is a lot of work. Requires:
  - lots of coding in C/C++
  - Much more emphasis on shader programming than in past offerings
  - Lots of math, linear algebra, matrices

Syllabus Summary

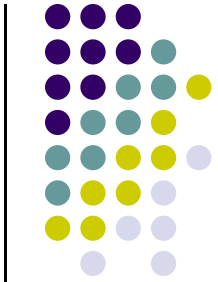


- 2 Exams (50%), 5 Projects (50%)
- Projects:
  - Develop OpenGL code on any platform, one ray tracing project
  - Final submission must run on CCC Linux machines
  - May discuss projects, turn in individual projects
- Class website: http://web.cs.wpi.edu/~emmanuel/courses/cs543/f11/
- Text:
  - Interactive Computer Graphics: A Top-Down Approach with Shader-based OpenGL by Angel and Shreiner (6th edition).
- Cheating: Immediate 'F' in the course
- Advice:
  - Come to class
  - Read the book
  - Understand concepts before coding

Computer Graphics Background



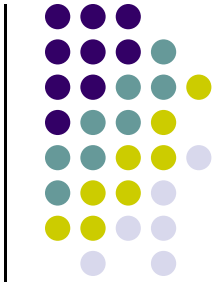
- Started early ’60s: Ivan Sutherland (MIT thesis)
- SIGGRAPH conference:
  - started 1969, about 30,000 annually. E.g. summer 2006: Boston
  - Attendees: artists, computer scientists, companies
- Computer Graphics has many aspects
  - Computer Scientists create graphics libraries, tools, packages (e.g. Maya)
  - Artists use CG tools/packages to create pretty pictures
  - Most hobbyists follow artist path. Not much math!



Computer Graphics Tools

- **CG tools:** hardware and software tools
- Hardware tools
  - Output devices: Video monitors, printers
  - Input devices:Mouse/trackball, pen/drawing tablet, keyboard
  - Graphics cards/accelerators (GPUs)
- Software tools (low level)
  - Operating system
  - Editor
  - Compiler
  - Debugger
  - Graphics Library (OpenGL)

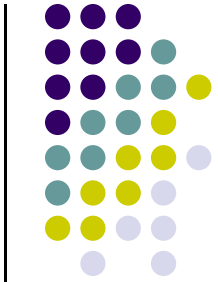
Graphics Processing Unit (GPU)



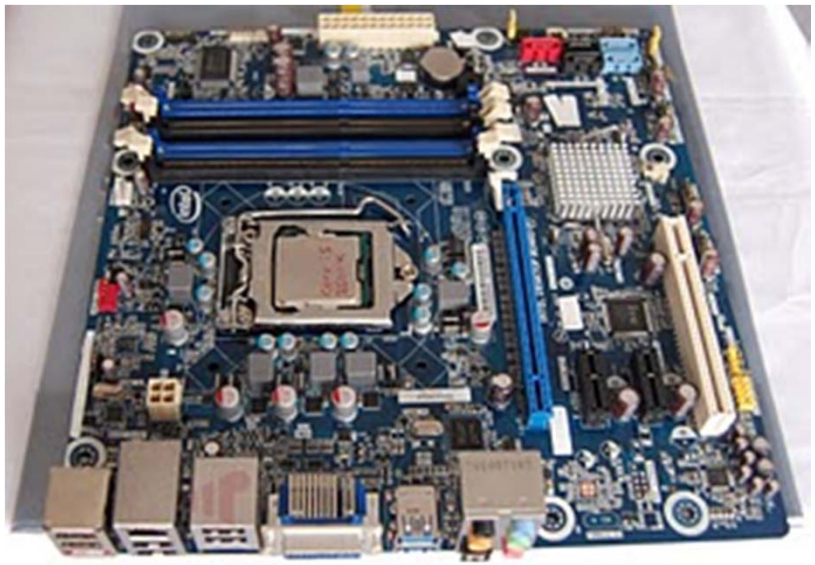
- Initially, just hardcode graphics library onto chip, increase speed
- Powerful, inexpensive, Giga-FLOPS arithmetic ability!
- **Programmable:** in last 8 years

  New operations just added. Possibility to apply to non-graphics application.
- Increasing precision
- Located either on the motherboard (Intel) or Separate graphics card (Nvidia or ATI)

Graphics Processing Unit (GPU)

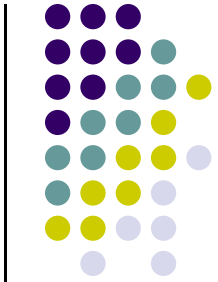


- Either on motherboard or separate card



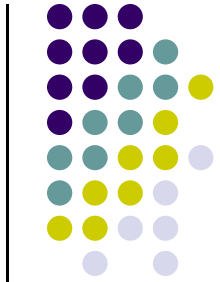
On motherboard

On separate card



Computer Graphics Libraries

- Functions/commands to draw line, circle, cube, etc
- Elaborate: pull-down menus, 3D coordinate system, etc
- Previously device-dependent
  - Different OS => different graphics library
  - Difficult to port (e.g. move program Windows to Linux)
  - Error Prone
- Now device-independent libraries
  - APIs: OpenGL, DirectX, java3D
  - Working OpenGL program easily moved from Windows to Linux, etc

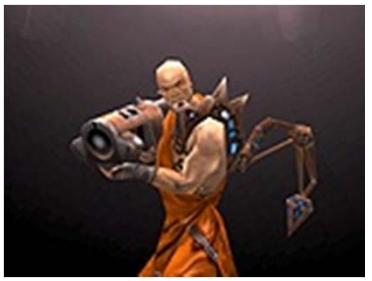


Motivation for CG

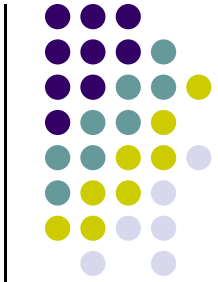
- Pretty pictures
- Humans respond better to pictures than text
- Reasons you are studying CG?
  - Better information presentation
  - Job in computer graphics (games, movies, etc)
  - Get a grade (one of required 4000 courses)??
  - Take advanced graphics or visualization course
  - Do research in graphics

Uses of Computer Graphics

- **Art, entertainment, publishing:**
  - movies, TV, books, magazines, games



Courtesy: Madden NFL game



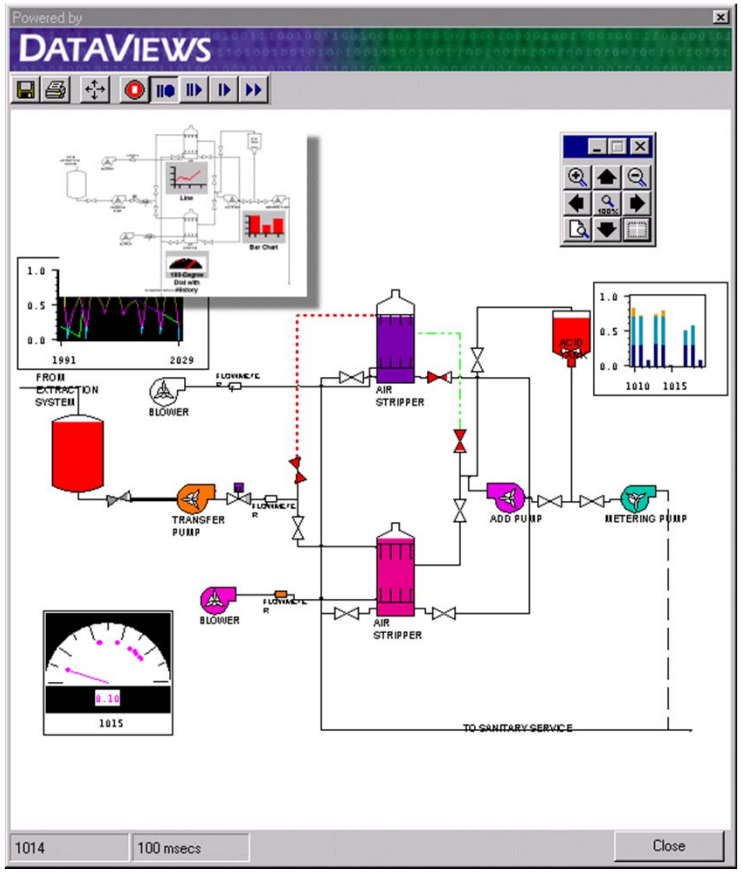
Uses of Computer Graphics

- **Image processing:**
  - alter images, remove noise, super-impose images

Courtesy: Forrest Gump movie

Uses of Computer Graphics

- **Process monitoring:**
  - Layout of large systems or plants



Courtesy:

Dataviews.de

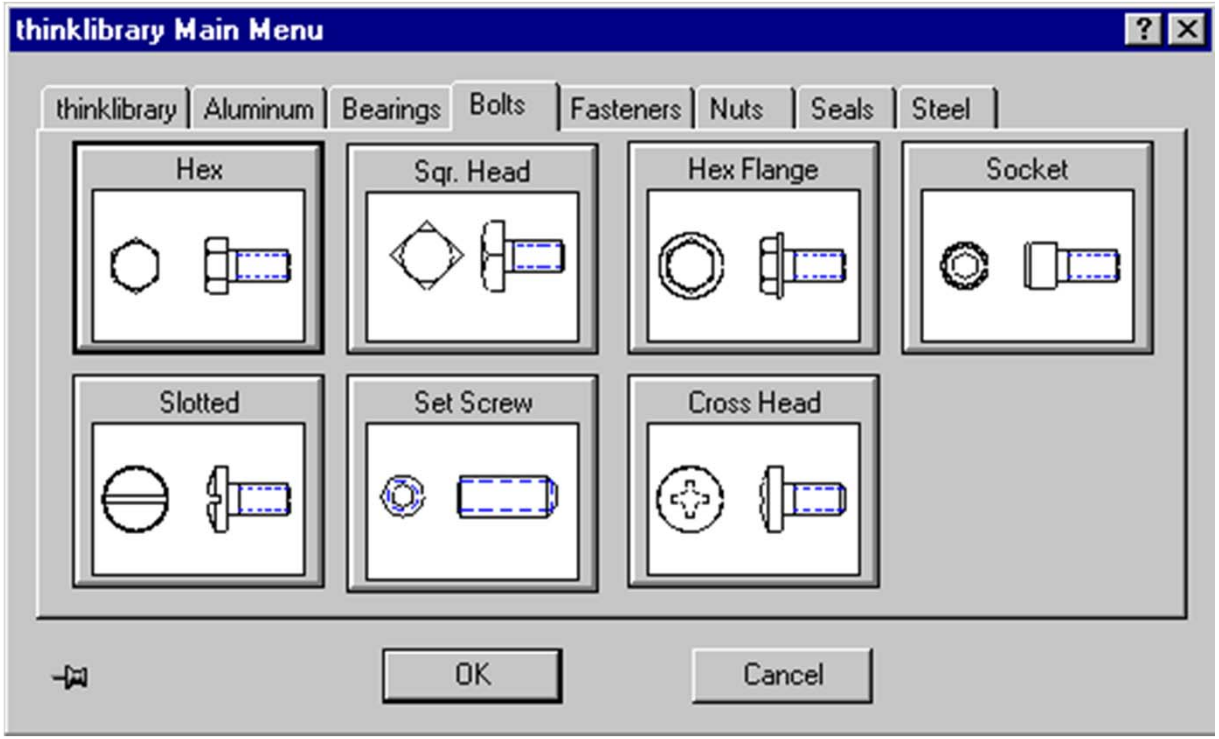
Uses of Computer Graphics

- **Display simulations:**
  - flight simulators, virtual worlds

Courtesy: Evans and Sutherland

Uses of Computer Graphics

- **Computer-aided design:**
  - architecture, electric circuit design

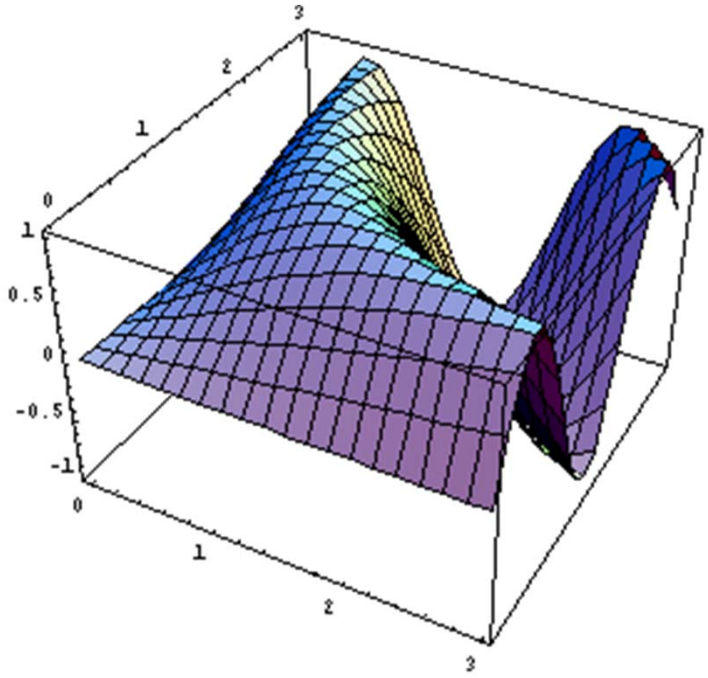


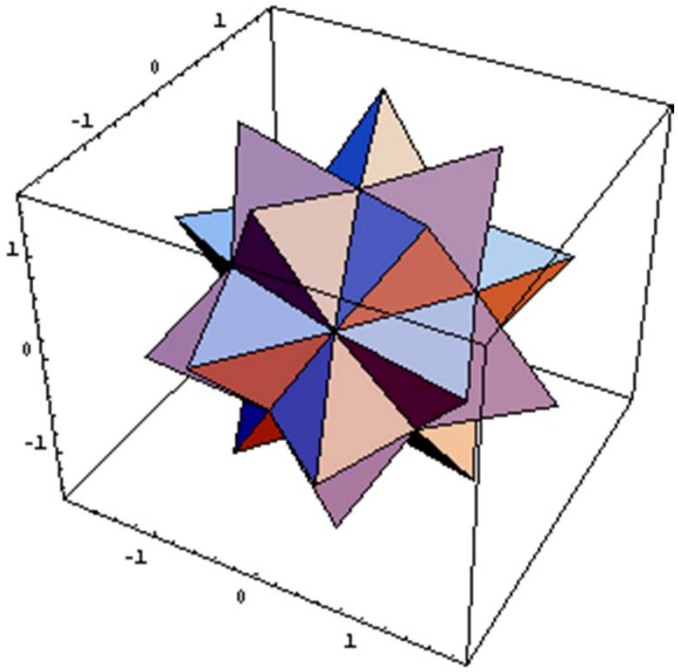
Courtesy:

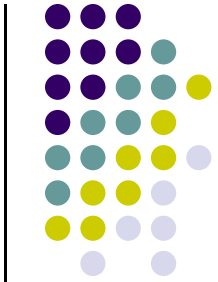
cadalog.com

Displaying Mathematical Functions

- E.g., Mathematica®

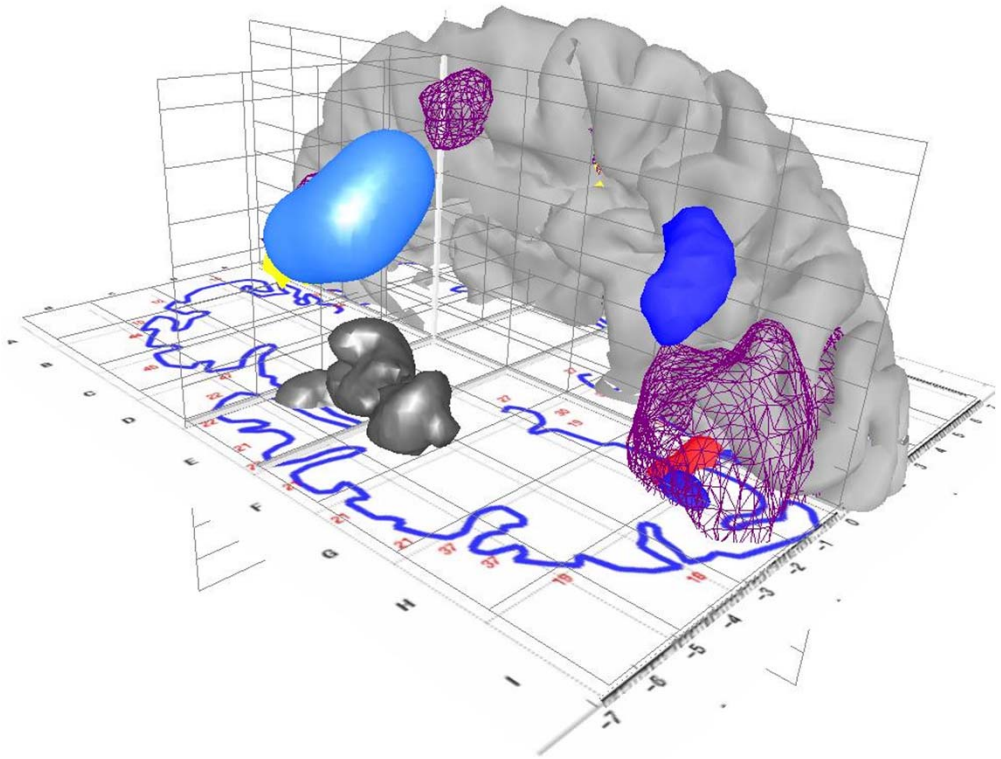






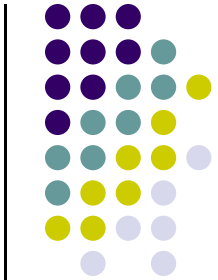
Uses of Computer Graphics

- **Scientific analysis and visualization:**
  - molecular biology, weather, matlab, Mandelbrot set



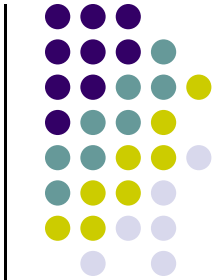
Courtesy:

Human Brain Project, Denmark



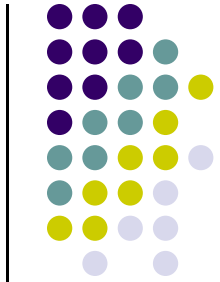
CG use example

- Biggest CG consumers today are
  - Movies (Hollywood): animated movies and special effects
  - Computer Games: e.g. Madden NFL Football 2011
  - Computer games now generate more billions than movies?
- Animated movies
  - Example: Toy Story 3 (Trailer starts at 0:48)
- Game trailer
  - Example: Final Fantasy XIV (Trailer starts at 0:25)



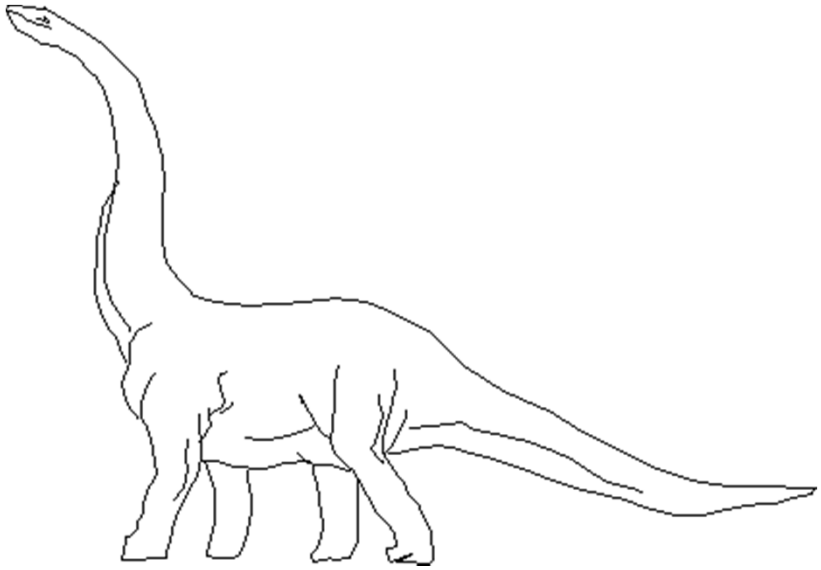
Elements of CG

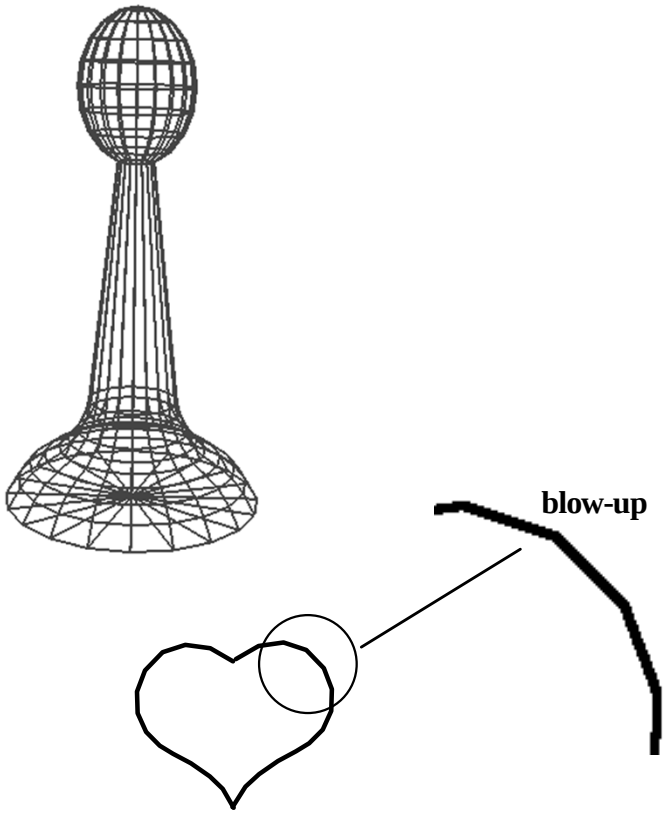
- **Polylines:** connected straight lines (edges, vertices)
- **Text:** font, typeface
- **Filled regions:** colors, patterns
- **Raster images:** pixels have values (pixmap)

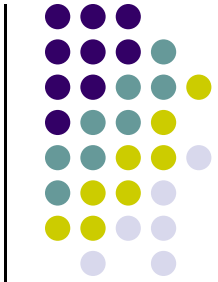


Polylines

- **Polyline:** connected sequence of straight lines



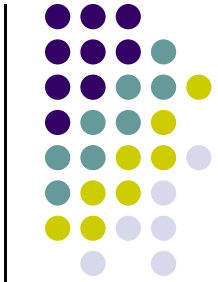




Polyline Attributes

- Color
- Thickness
- Stippling of edges (dash pattern)

Text



- Devices have:
  - **text mode**
  - **graphics mode**.
- **Text mode:** Text uses built-in character generator
- **Graphics mode:** Text is drawn
- **Text attributes:** Font, color, size, spacing, and orientation.

Big Text

Little Text

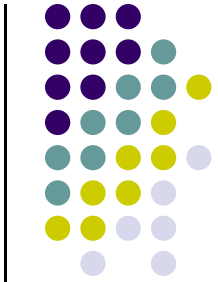
Shadow Text

Distorted text

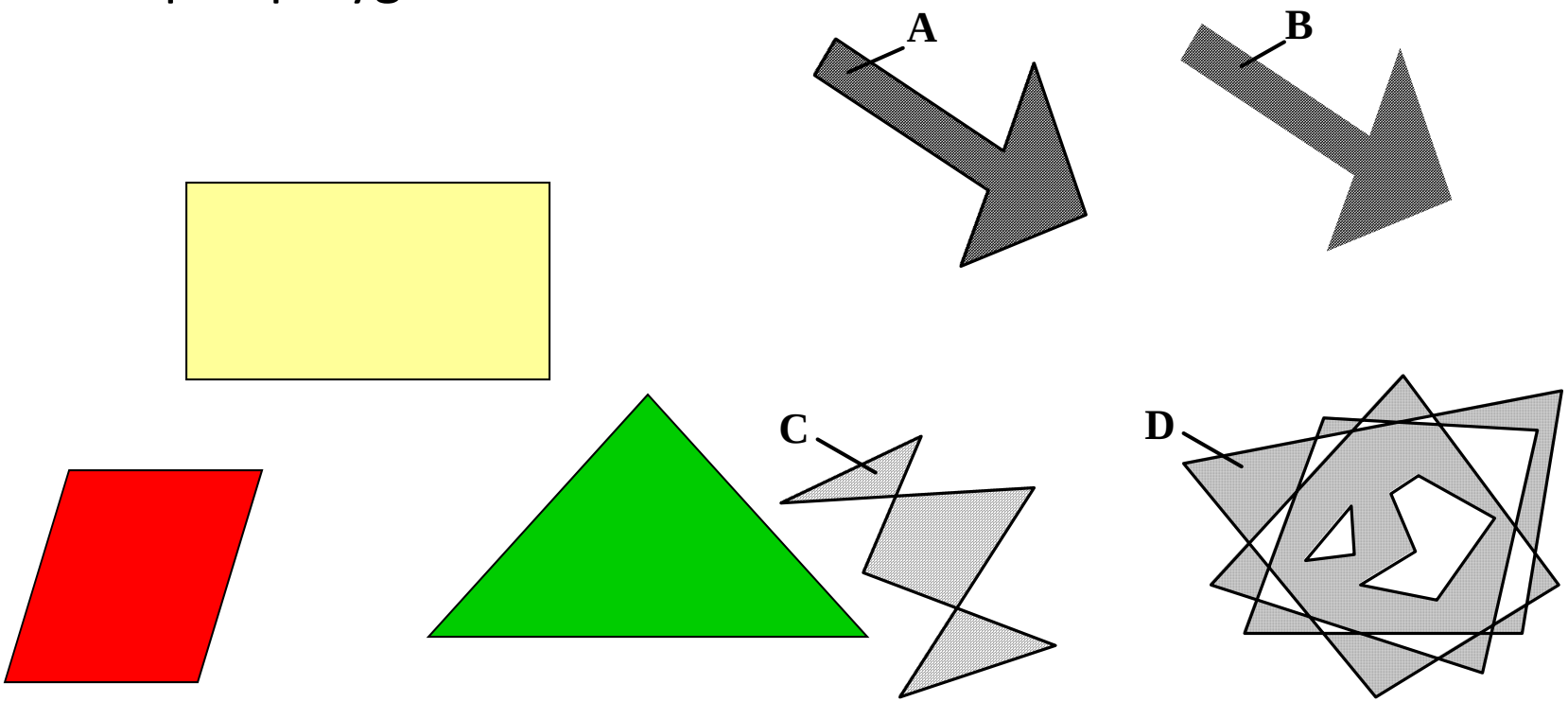
Rotated Text **Outlined text**

SMALLCAPS

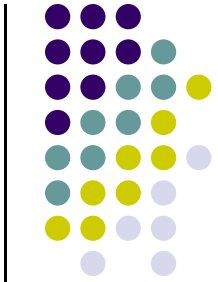
Filled Regions



- **Filled region** primitive is a shape filled with some color or pattern
- Example: polygons

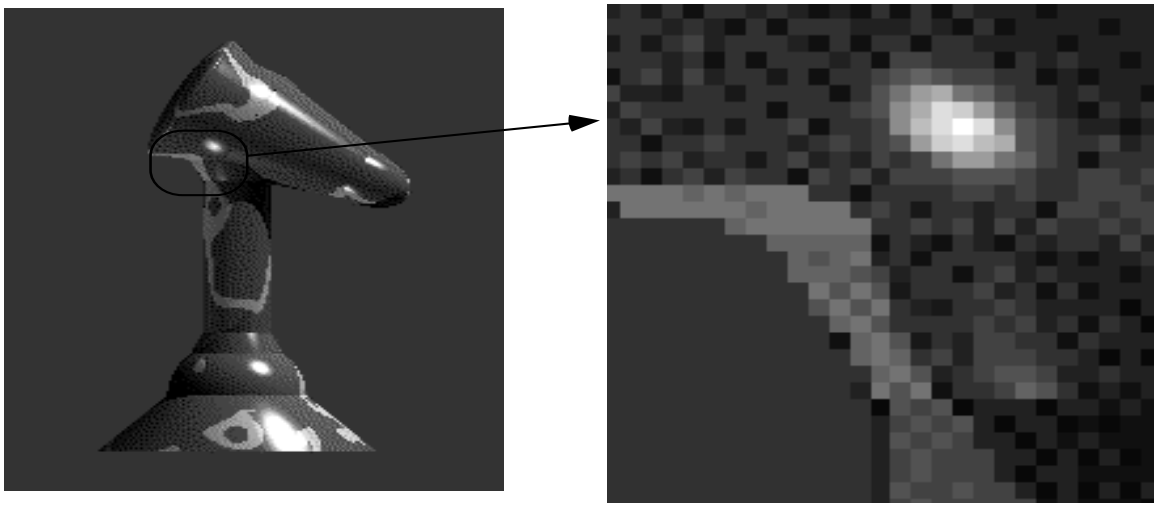


Raster Images



- Raster image (picture) is made up of many small cells (pixels, for “picture elements”), in different colors or grayscale.

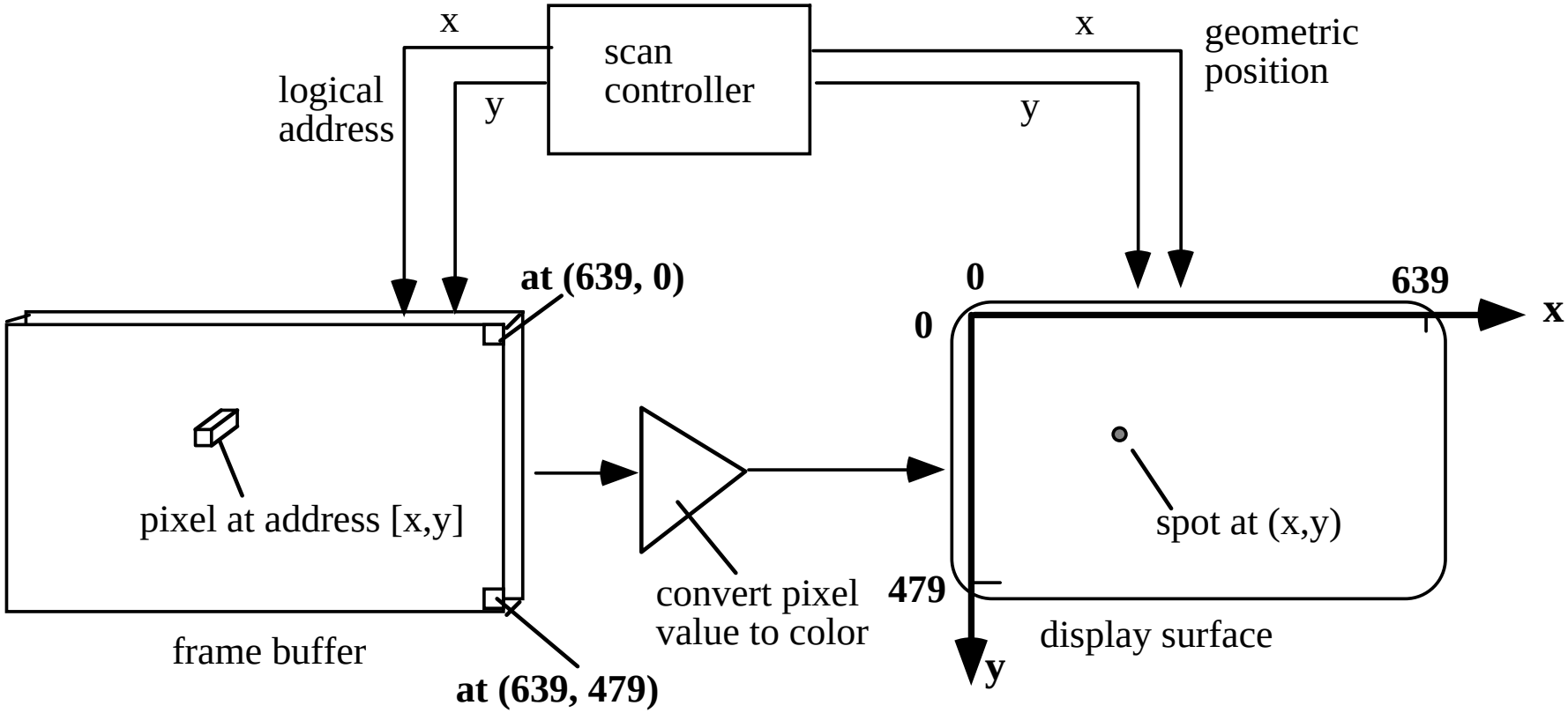
(Right: magnified image showing pixels.)



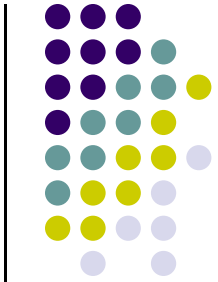
Framebuffer

Dedicated memory location:

- draw in framebuffer shows up on screen
- Located either on CPU or GPU



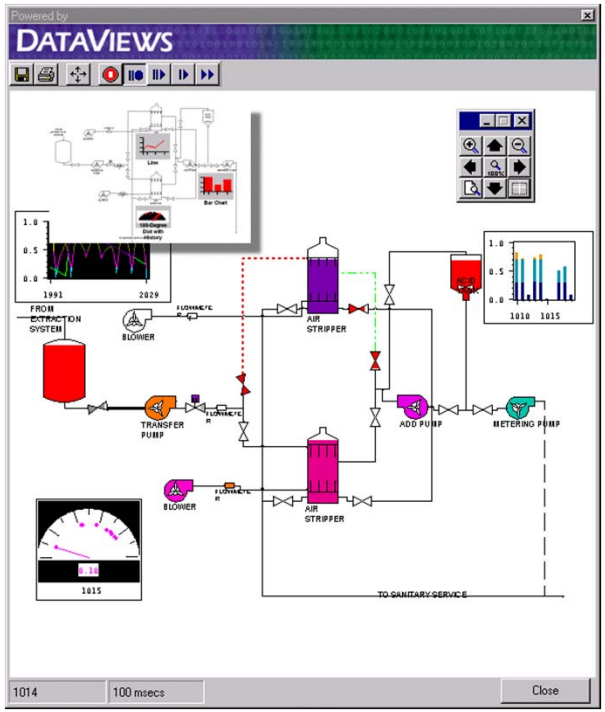
Types of Input Devices



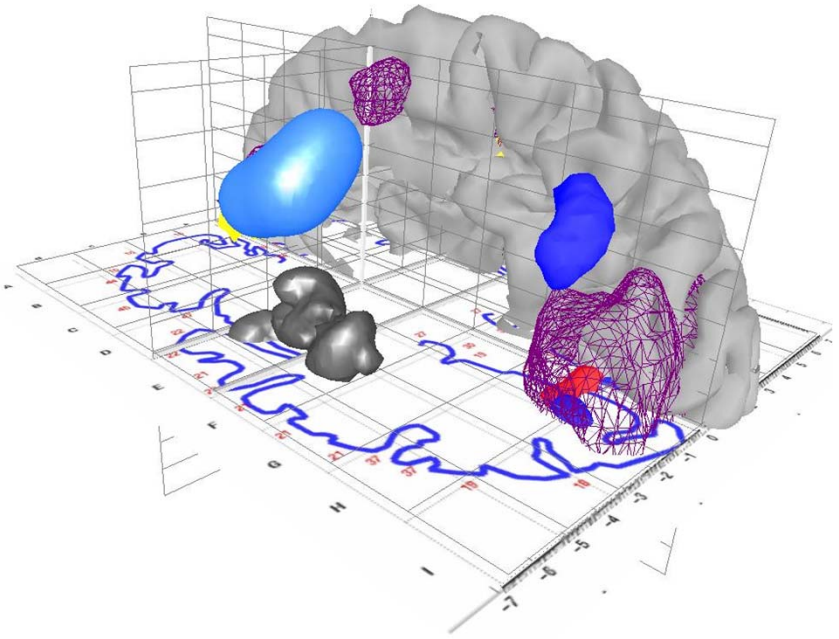
- **String:** produces string of characters. (e.g keyboard)
- **Valuator:** generates number between 0 and 1.0 (e.g. knob)
- **Locator:** User points to position on display (e.g. mouse)
- **Pick:** User selects location on screen (e.g. touch screen in restaurant, ATM)

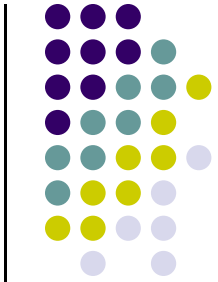
2D Vs. 3D

- 2D:
  - Flat
  - (x,y) color values on screen
  - Objects no depth or distance from viewer



- 3D
  - (x,y,z) values on screen
  - Perspective: objects have distances from viewer

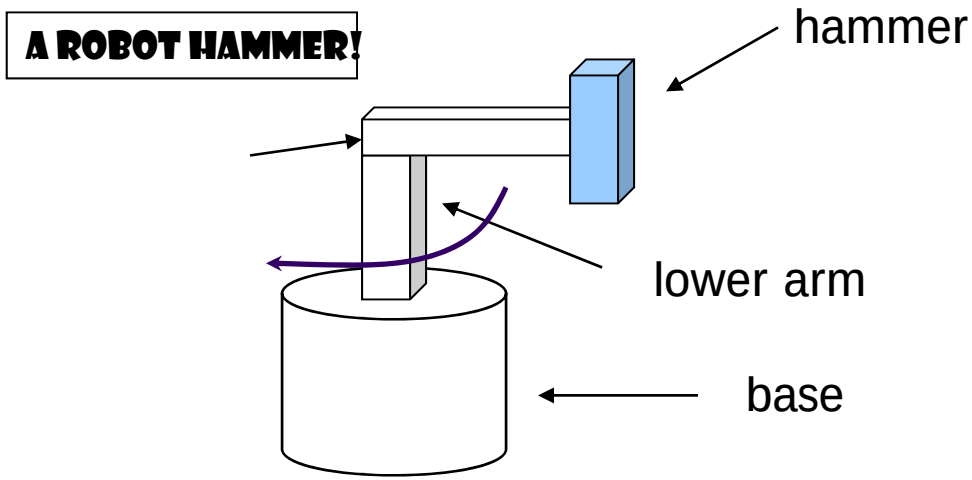




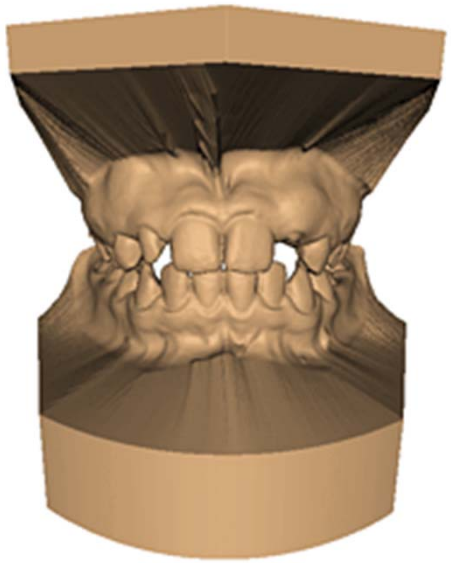
Creating 3D

- Start with 3D shapes (modeling)
  - Basic shapes(cube, sphere, etc), meshes, etc
  - Scale them (may also stretch them)
  - Position them (rotate them, translate, etc)
- Then, add 3D effects to make scene look real
  - Color and shading
  - Shadows
  - Texture mapping
  - Fog
  - Transparency and blending
  - Anti-aliasing
- Practical note: software packages for modeling and rendering can be purchased (Maya, 3D studio max, etc)

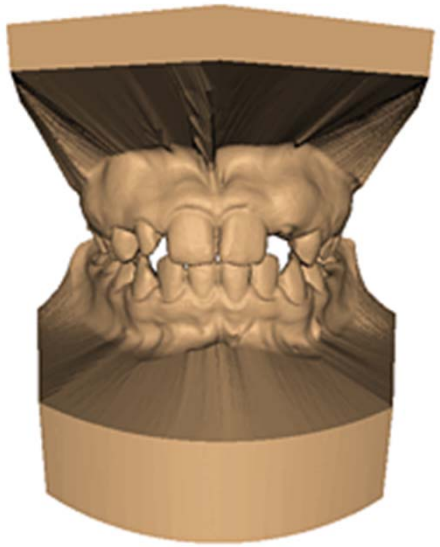
3D Modeling example: Robot Hammer



3D Modeling example: Polygonal Mesh



Original: 424,000 triangles

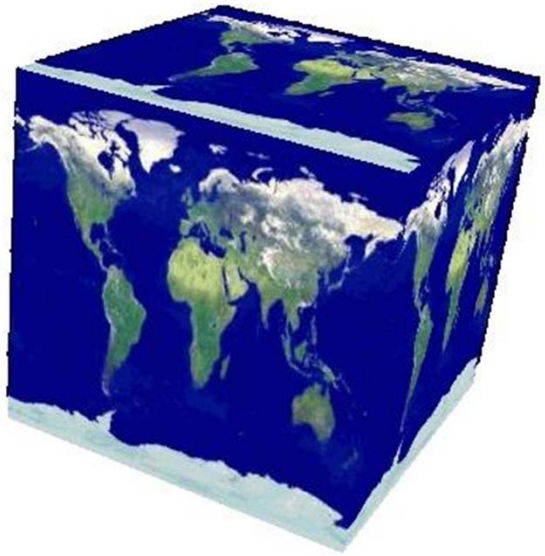


60,000 triangles (14%).

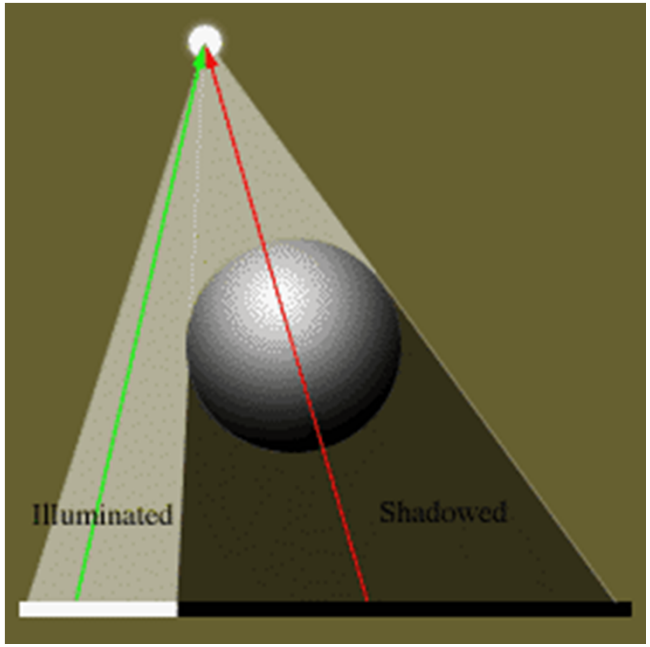
1000 triangles (0.2%)

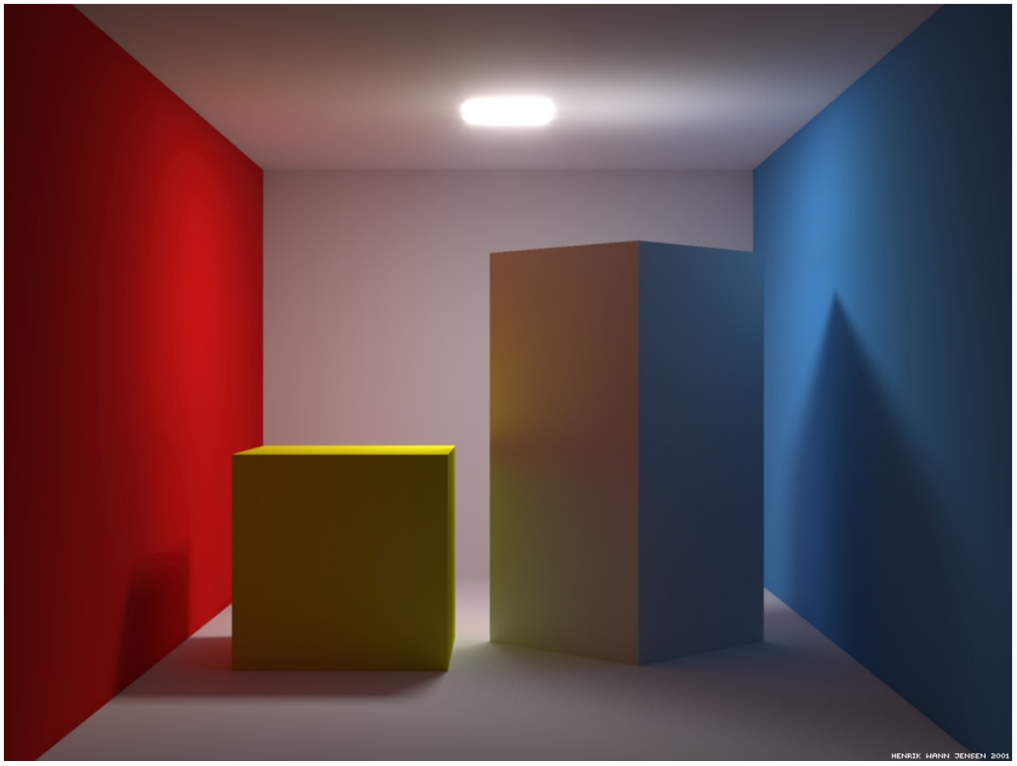
(courtesy of Michael Garland and Data courtesy of Iris Development.)

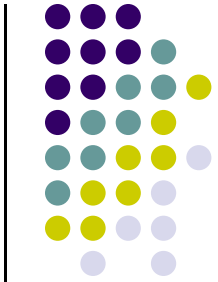
3D Effects example: Texturing



3D Effects example: Shadows







References

- Angel and Shreiner, Interactive Computer Graphics (6th edition)
- Hill and Kelley, Computer Graphics using OpenGL (3rd edition), Chapter 1