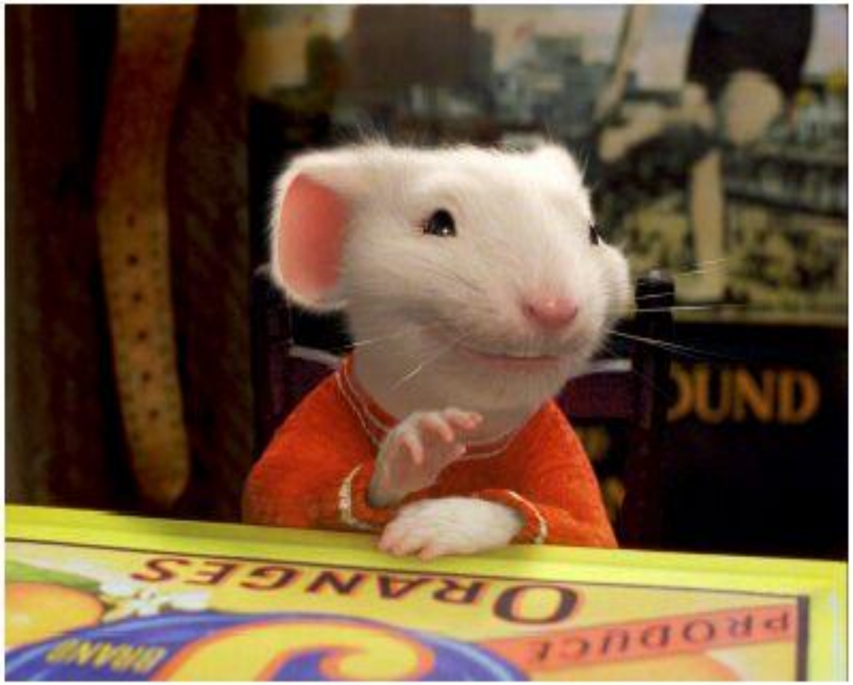


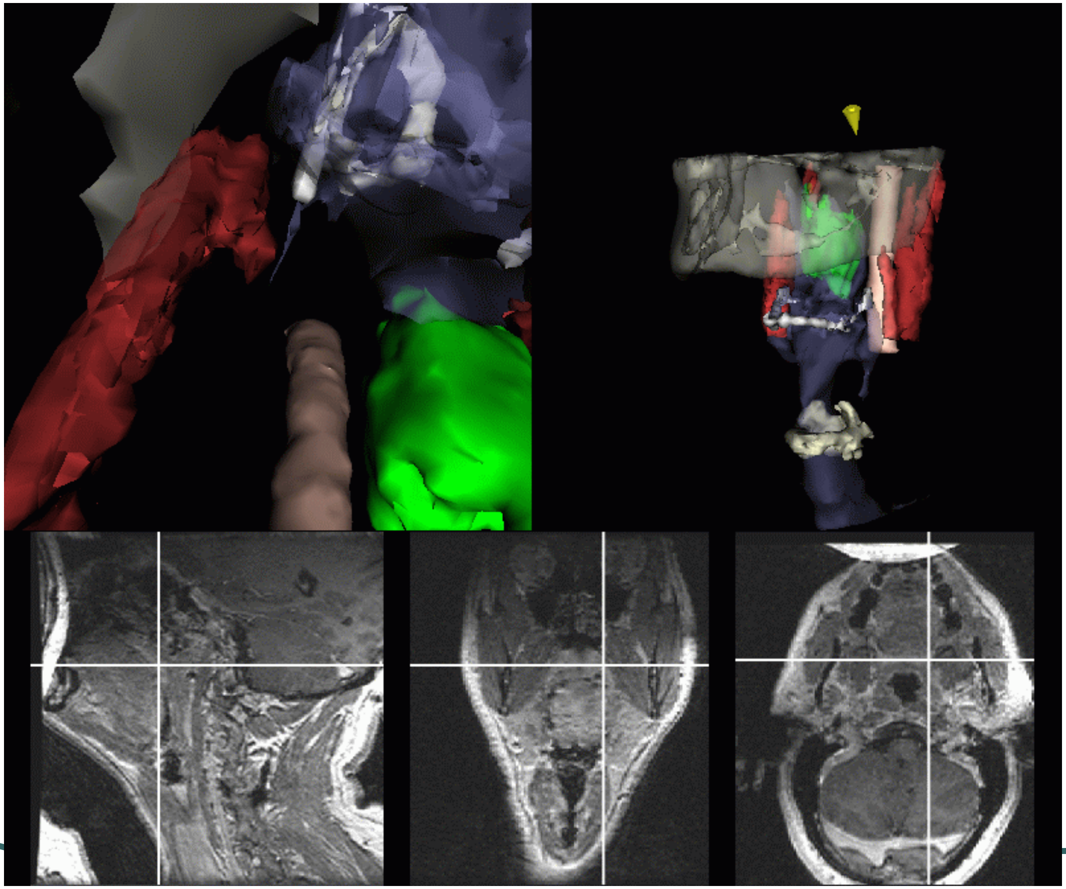
Introduction to Computer Graphics

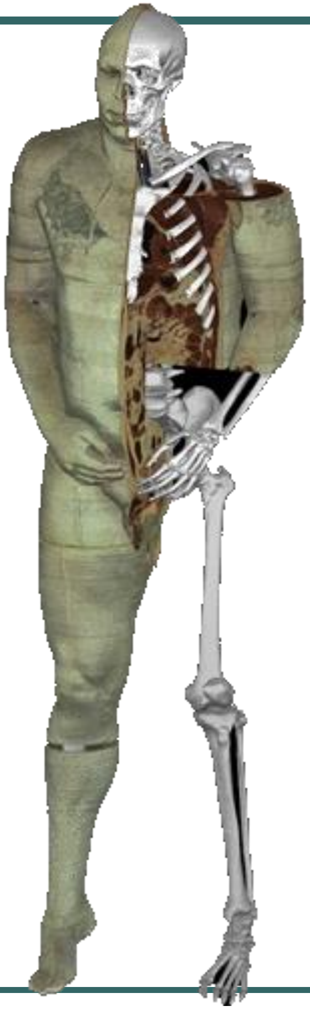
Prerequisites

- You will be writing programs
  - Non-trivial data structures, pointers
- An ability to learn a programming library on your own
  - OpenGL
- Comfortable with matrix algebra and calculus
  - Basic linear algebra used

Graphics Applications

- Medical Visualization



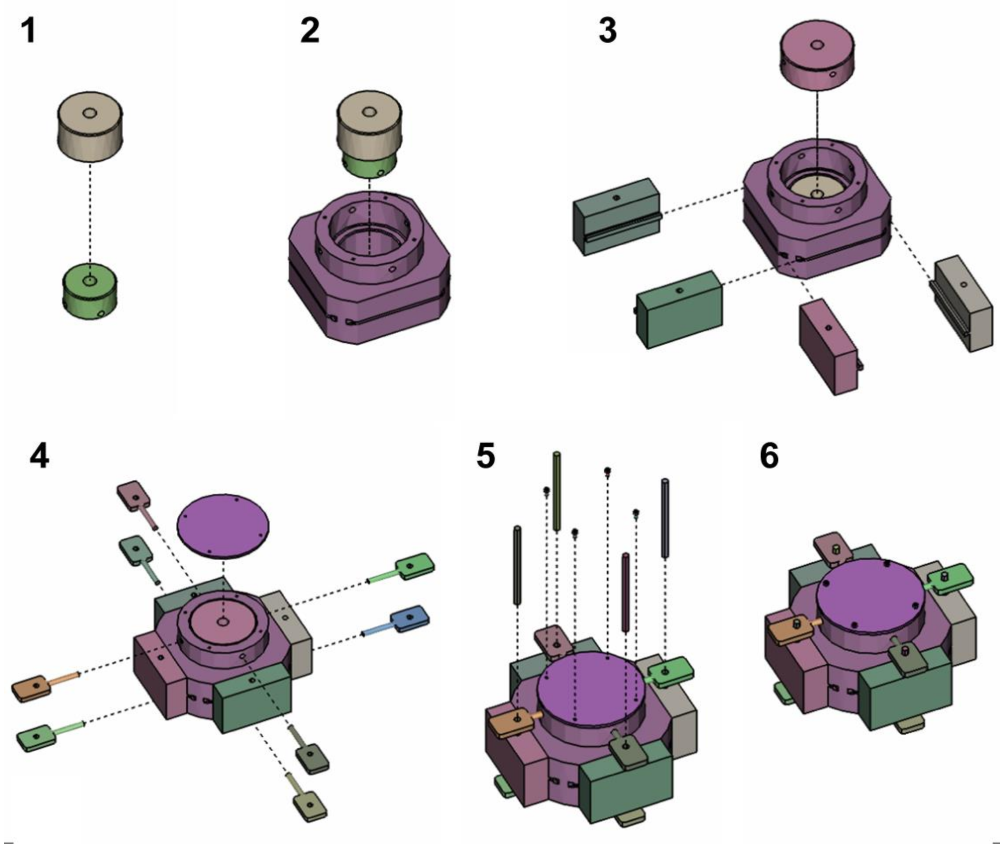


Graphics Applications

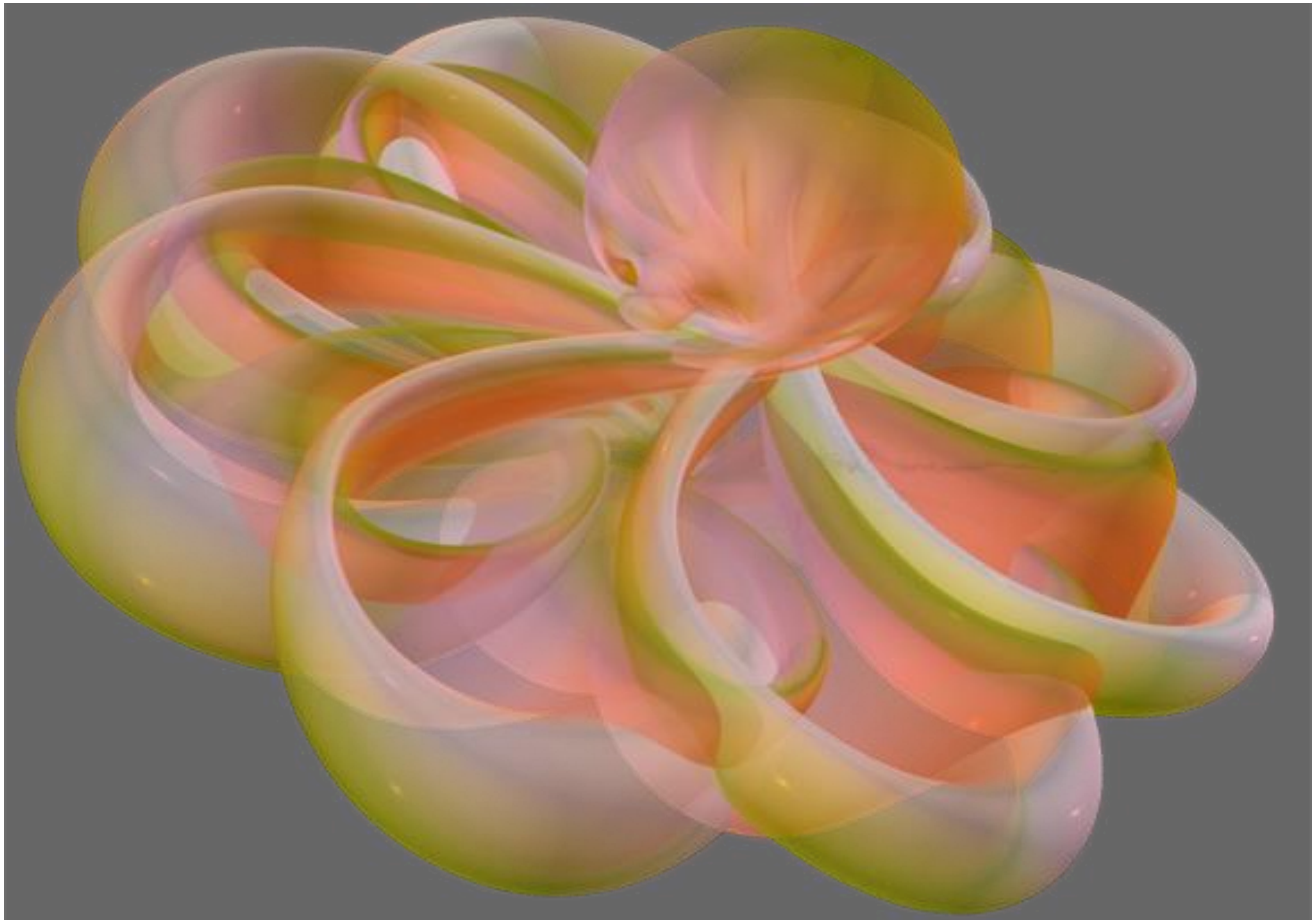
- Entertainment: Cinema

Graphics Applications

- Training

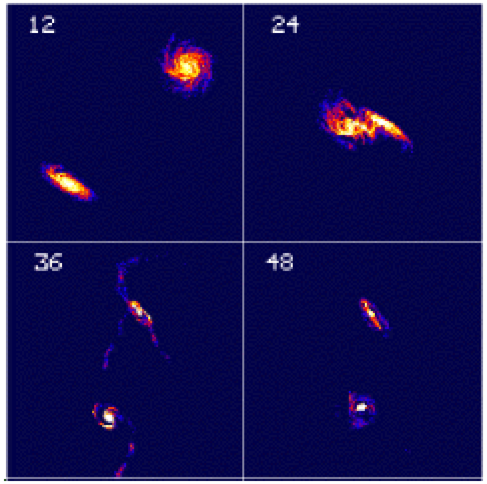


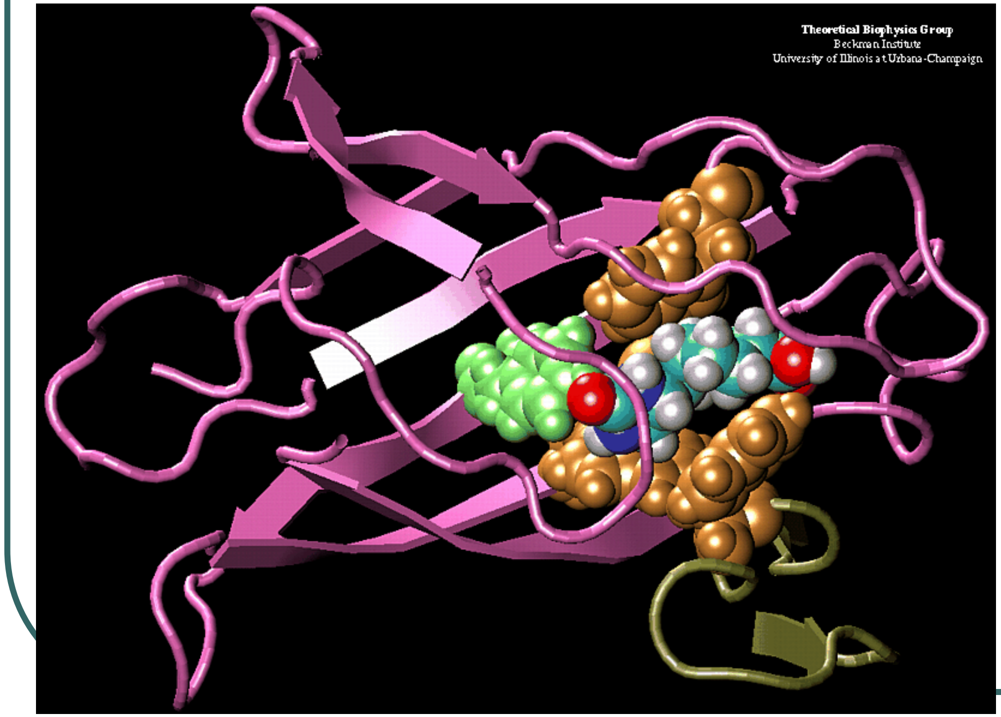
Education



Graphics Applications

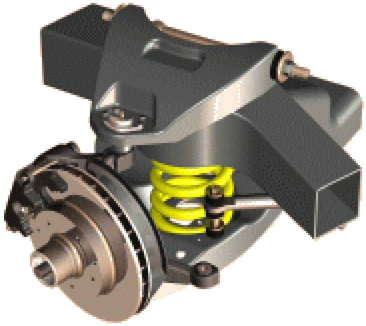
- Scientific Visualization

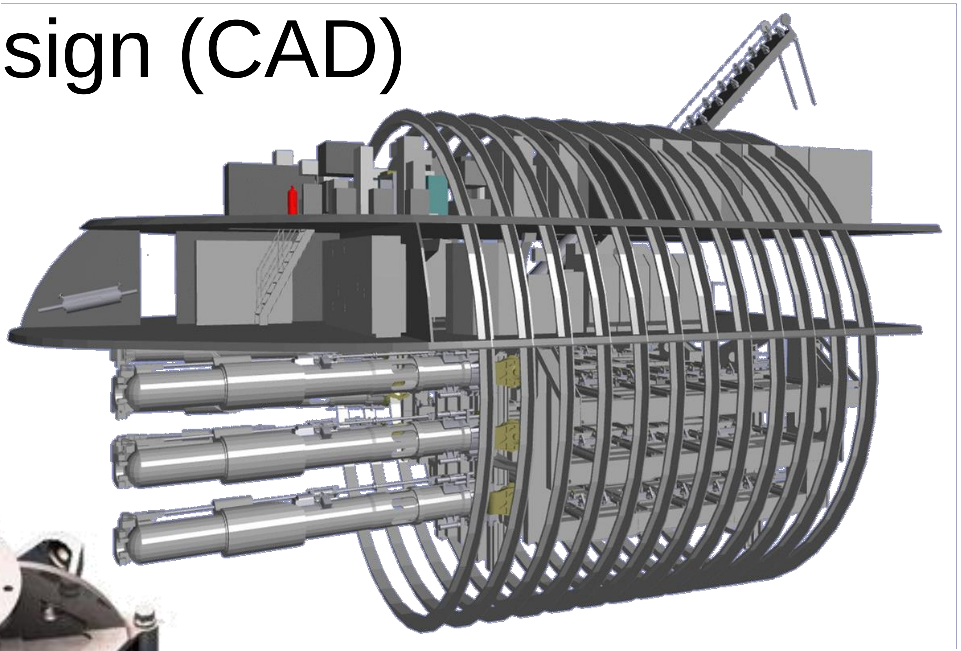


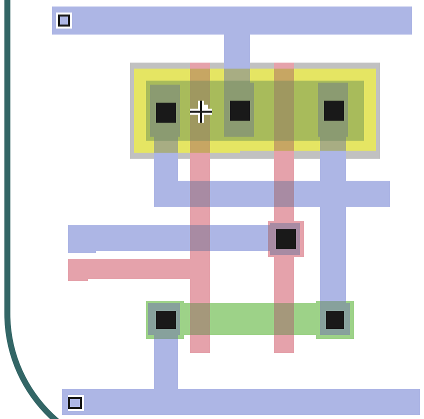

Theoretical Biophysics Group
Beckman Institute
University of Illinois at Urbana-Champaign

Graphics Applications

- Computer Aided Design (CAD)







Graphics Applications

- Entertainment: Games

Polyphony Digital: Gran Turismo 3, A Spec

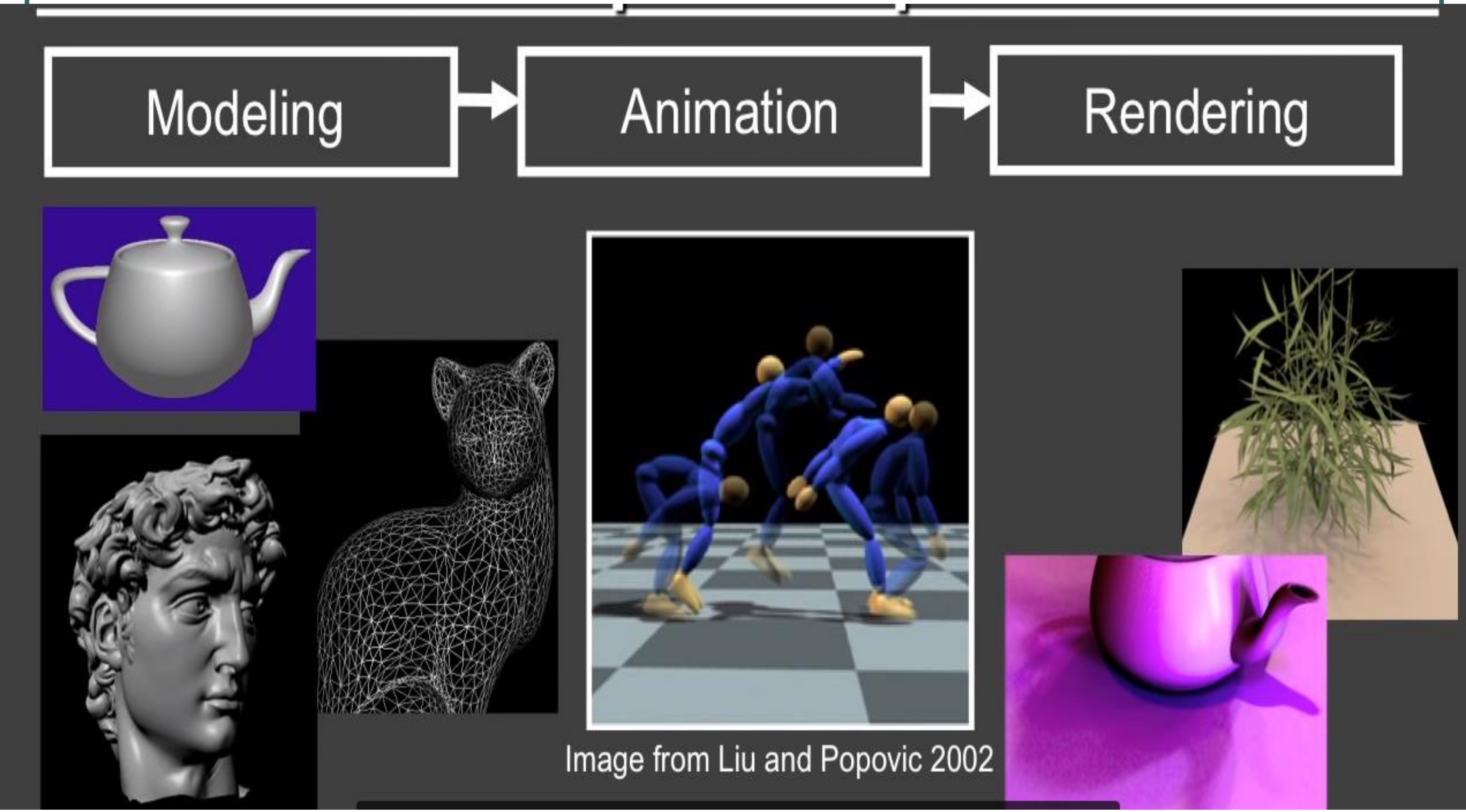
What is Computer Graphics ?

- The term of Computer Graphics includes almost everything on computers that is not text or sound.
- It is creation, storage and manipulation of images and models using Algorithms and data structures.
- It is art of drawing pictures on computer using programming

Computer Graphics Made Up of 3Components

- Image:
  - Image is a combination of pixels, visual representation of something.
- 1- Model:
  - (shape) creating and representing the geometry of objects in the 3D world.
- 2- Animation:
  - (movement) describing how objects change in time
- 3- Rendering:
  - (light, perspective) generating 2D images of the objects

Computer Graphics Made Up of 3 Components



Modeling → Animation → Rendering

Image from Liu and Popovic 2002

Why Study Computer Graphics?

- Graphics is cool
  - I like to see what I’m doing
  - I like to show people what I’m doing
- Graphics is interesting
  - Involves *simulation*, AI, *algorithms*, *architecture*…
- Graphics is fun

Introduction to OpenGL

OpenGL: Setup in Windows

- You can download this library from http://www.xmission.com/~nate/glut/glut-3.7.6-bin.zip
- Assume that Visual Studio was installed at the directory “C:\Program Files\Microsoft Visual Studio 10.0\”.

- Extract *glut-3.7.6-bin.zip* and place its contents as follows:

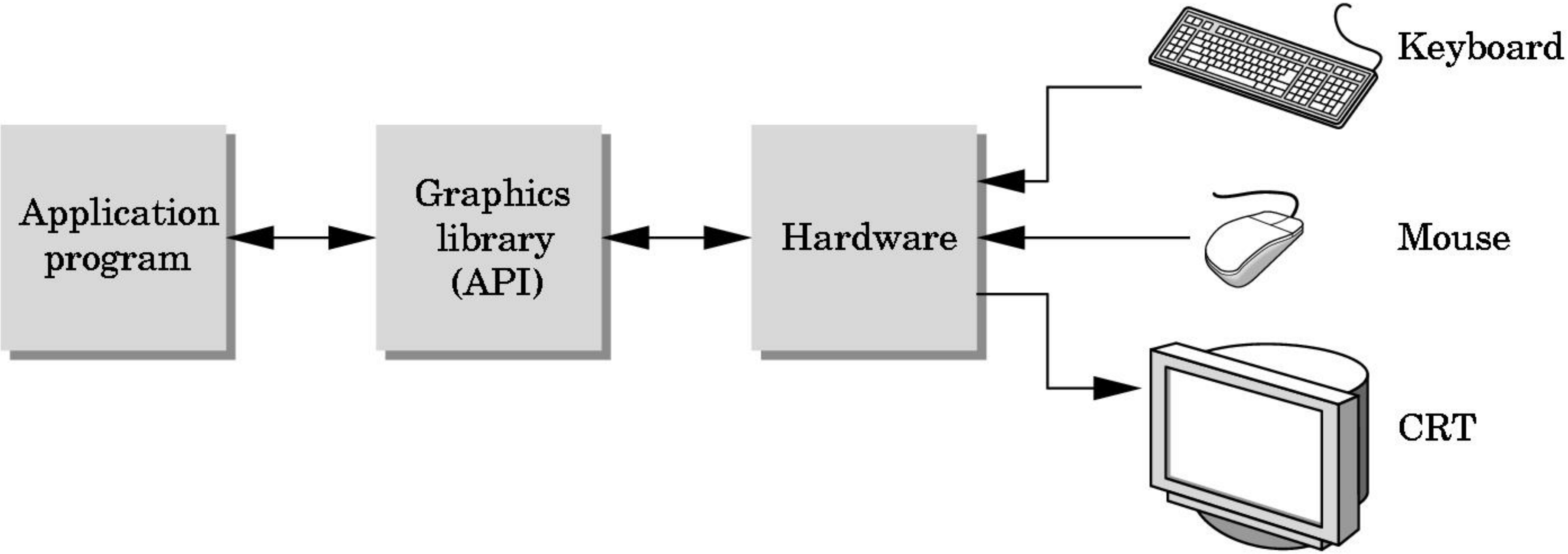
- Put *glut.h* inside “C:\Program Files (x86)\Microsoft Visual Studio 10.0\VC\include\gl\” (you may need to create the directory **gl** yourself).
- Put *glut32.lib* inside “C:\Program Files (x86)\Microsoft Visual Studio 10.0\VC\lib\”.
- **Windows 32-Bit Users**: Put *glut32.dll* inside “C:\Windows\System32\”.
- **Windows 64-Bit Users**: Put *glut32.dll* inside “C:\Windows\SysWOW64\”.

What is Opengl

- OpenGL is a software interface to graphics hardware.

The Programmer's Interface

- Programmer sees the graphics system through a software interface: the Application Programmer Interface (API).



API Contents

- Functions that specify what we need to form an image
  - Objects
  - Viewer
  - Light Source(s)
  - Materials
- Other information
  - Input from devices such as mouse and keyboard
  - Capabilities of system

History of OpenGL

- Silicon Graphics (SGI) revolutionized the graphics workstation by implementing the pipeline in hardware (1982)
- To access the system, application programmers used a library called GL
- With GL, it was relatively simple to program three dimensional interactive applications

OpenGL: What is It?

- The success of GL lead to OpenGL (1992), a platform-independent API that was
  - Easy to use
  - Close enough to the hardware to get excellent performance
  - Focus on rendering
  - Omitted windowing and input to avoid window system dependencies

OpenGL Evolution

- Controlled by an Architectural Review Board (ARB)
  - Members include SGI, Microsoft, Nvidia, HP, 3DLabs, IBM,…….
  - Relatively stable (present version 2.0)
    - Evolution reflects new hardware capabilities
      - **3D texture mapping and texture objects**
      - **Vertex programs**
  - Allows for platform specific features through extensions

Why Opengl

1. **Open source graphics library**

   http://www.opengl.org

2. **Device-independent graphics programming**

3. **Windows based programming**

   - Event- driven programming,

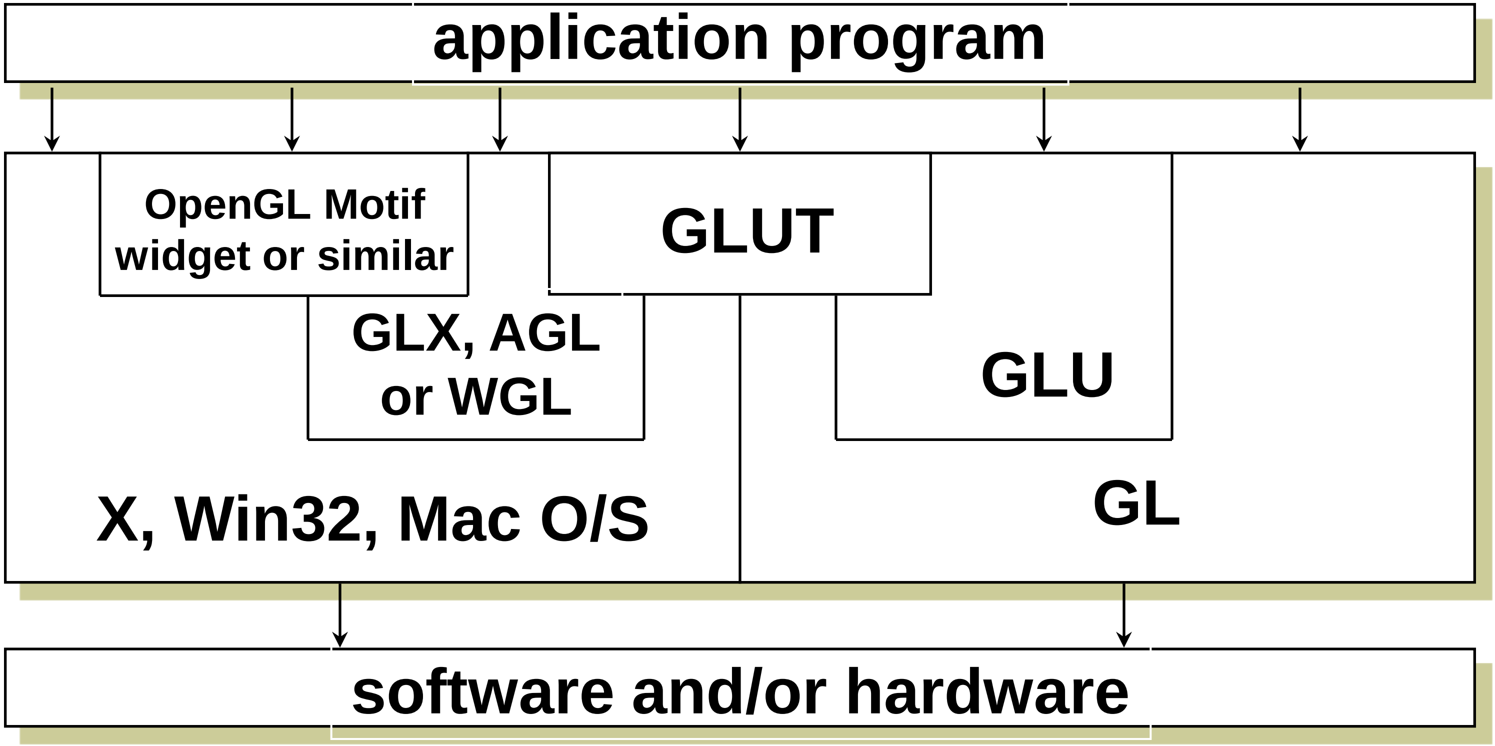
     event queue, Callback functions, Event event loop.

   - Glut library support event

OpenGL Libraries

- **GL** (**G**raphics **L**ibrary): Library of 2-D, 3-D drawing primitives and operations
  - API for 3-D hardware acceleration
- **GLU** (**GL U**tilities): Miscellaneous functions dealing with camera set-up and higher-level shape descriptions
- **GLUT** (**GL U**tility **T**oolkit): Window-system independent toolkit with numerous utility functions, mostly dealing with user interface

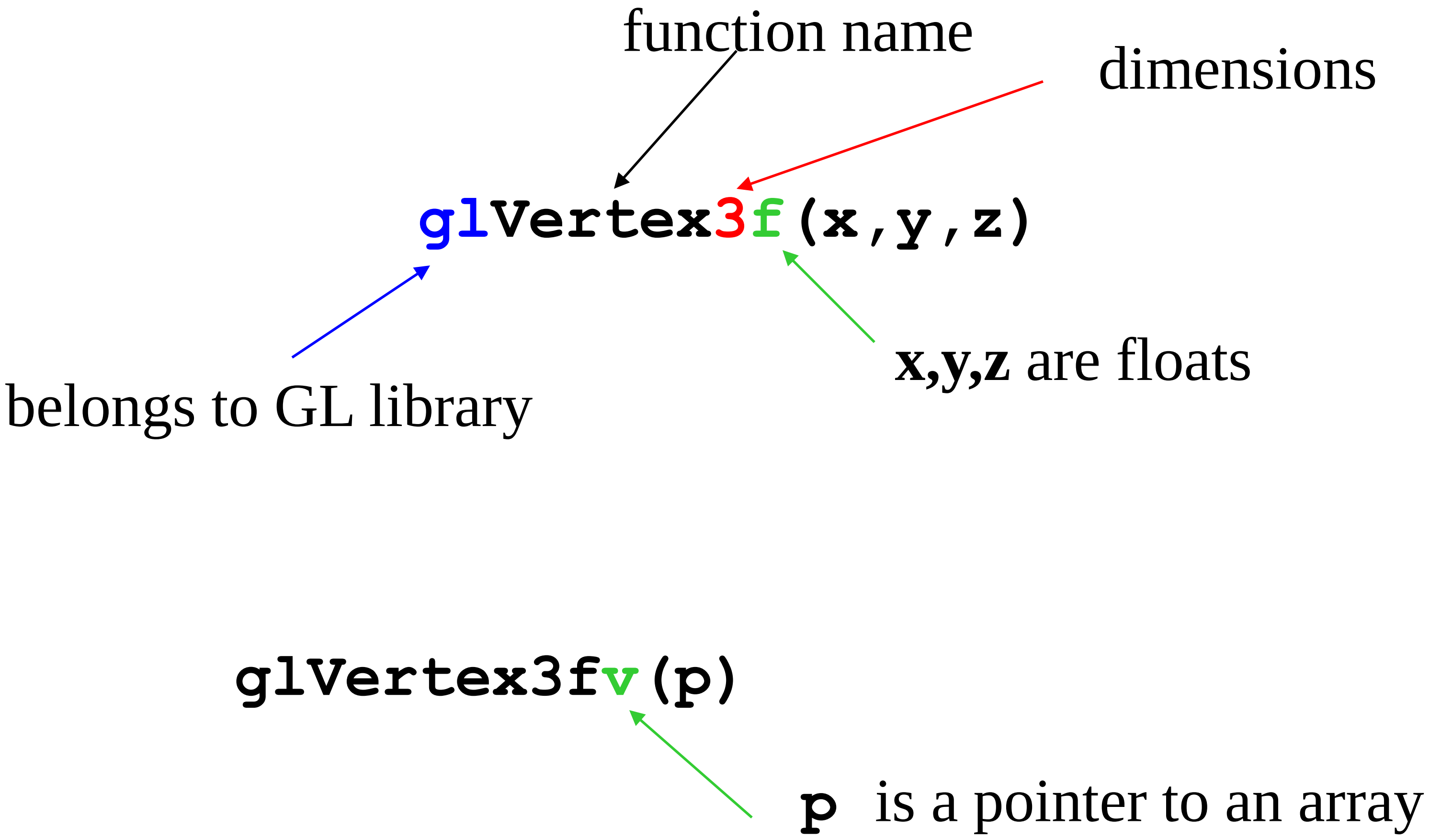
Software Organization



Lack of Object Orientation

- OpenGL is not object oriented so that there are multiple functions for a given logical function
    - **`glVertex3f`**
    - **`glVertex2i`**
    - **`glVertex3dv`**
- Underlying storage mode is the same
- Easy to create overloaded functions in C++ but issue is efficiency

OpenGL function format



function name

dimensions

glVertex3f(x,y,z)

belongs to GL library

x,y,z are floats

glVertex3fv(p)

p is a pointer to an array

Program Structure

- include files
- Main function
  1. Initialize glut and display
  2. Create window
  3. Register Callbacks functions
  4. Additional initialization for creating coordinate system
  5. Main Loop (event loop)

Program Structure

- Registered functions
  - Display function(rendering func), Mouse, Keyboard, Motion, Redisplay ……functions

simple.c revisited

```
#include <GL/glut.h>

int main(int argc, char** argv)
{
   glutInit(&argc,argv);
   glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
   glutInitWindowSize(500,500);
   glutInitWindowPosition(0,0);
   glutCreateWindow("simple");
   glutDisplayFunc(mydisplay);

   init();

   glutMainLoop();
}
```

includes **gl.h**

define window properties

display callback

set OpenGL state

enter event loop

GLUT functions

- **`glutInit`** allows application to get command line arguments and initializes system
- **`gluInitDisplayMode`** requests properties for the window (the *rendering context*)
  - RGB color
  - Single buffering
  - Properties logically ORed together
- **`glutWindowSize`** in pixels
- **`glutWindowPosition`** from top-left corner of display
- **`glutCreateWindow`** create window with title “simple”
- **`glutDisplayFunc`** display callback
- **`glutMainLoop`** enter infinite event loop