What is OpenGL?

- **A low-level graphics programming language**
  - Intended for interactive graphics: 2D & 3D
- **A software interface to graphics hardware**
- **Descendent of GL**
  - Cynics have called it GL with longer variable names and no window handling.
- **Implementations**
  - For the Sparcs and Linux PCs we have Mesa, a freeware implementation
  - SGIs have SGI's implementation. ~ 1500% faster.

How does it work?

- **A State Machine**
  - Set variables in the state, such as color, current viewing position, line width, material properties...
  - These variables then apply to every subsequent drawing command
  - State variables have default values
- **What it covers**
  - OpenGL's **Primitives** are points, lines and polygons
  - Doesn't do windowing. Use Xforms (FLTK?)...

Setting up the window

- **The coordinate system**
  - glOrtho(left, right, bottom, top, near, far);
    - e.g., glOrtho(0, 100, 0, 100, -1, 1);
  - For now, near & far should always be -1 & 1
- **Clearing the screen**
  - glClearColor(r, g, b, a);
    - a is the alpha channel; set this to 0.
  - glClear(GL_COLOR_BUFFER_BIT);
    - glClear can clear other buffers as well, but we're only using the color buffer...

Setting the Color

- **glColor3f(r, g, b);**
  - All subsequent primitives will be this color.
  - Red, green & blue color model
  - Components are 0-1
- **Side-note**
  - OpenGL naming convention is:
    - gl<Command>[234][sifd][v] (args...)
  - s - short, i - integer, f - float, d- double
  - v - pointer to an array

Drawing a polygon

- **Put GL into draw-polygon state**
  glBegin(GL_POLYGON);
- **Send it the points making up the polygon**
  glVertex2f(x0, y0);
  glVertex2f(x1, y1);
  glVertex2f(x2, y2) ...
- **Tell it we're finished**
  glEnd();
- **That's it.**

Special Tricks

- **In place of GL_POLYGON:**
  GL_POINTS: plot points
  GL_LINES: draw lines
  GL_LINE_LOOP: framed polygon
- **Gouraud Shading:**
  Change the color between setting each vertex, and GL will smooth-shade between the different vertex colors.
- **Flushing the pipeline**
  glFlush();

Drawing A Box

```
void DrawBox()
{
    MakeWindow("Box", 400, 400);

    glOrtho(-1, 1, -1, 1, -1, 1);

    glClearColor(0.5, 0.5, 0.5, 1);
    glClear(GL_COLOR_BUFFER_BIT);

    glColor3f(1.0, 0.0, 0.0);

    glBegin(GL_POLYGON);
    /* or GL_LINES or GL_POINTS... */

    glVertex2f(-0.5, -0.5);
    glVertex2f( 0.5, -0.5);
    glVertex2f( 0.5,  0.5);
    glVertex2f(-0.5,  0.5);

    glEnd();
}
```


Miscellaneous

- **Example Code**
  We will give you code for a skeleton program that makes a window with an OpenGL canvas and a couple of buttons. It draws, and handles mouse events on the canvas.
- **Man pages**
  Html-ified OpenGL man pages are on the course software page. SGIs also have "online books" which include OpenGL documentation. We also have the whole xforms manual online , html-ified.