FLTK

- C++ graphical user interface toolkit
- Provides similar functionality as those window classes (CWnd, CButton, etc.) found in MFC.

Naming

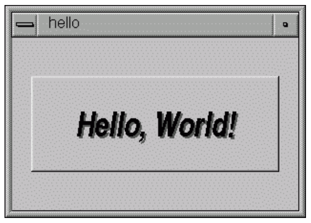
- Function names: `Fl::foo()` or `fl_foo()`
- Class and type names: `Fl_foo`
- Constant enumeration: `FL_FOO`
- Header files: `<FL/foo.h>`

Simple Program

```
#include <FL/Fl.h>
#include <FL/Fl_Window.H>
#include <FL/Fl_Box.h>

int main(int argc, char *argv) {
  FL_Window *w = new Fl_Window(300, 180);
    FL_Box *b = new Fl_Box(20, 40,
      260, 100,
      "Hello World!");
    b->(FL_UP_BOX);
    b->labelsize(36);
    b->labelfont(FL_BOLD);
    b->labeltype(FL_SHADOW_LABEL);
  w->end();
  w->show(argc, argv);
  return Fl::run()
}
```

Simple Program (Cont'd)



OpenGL

- 2D/3D graphics library by Silicon Graphics Inc. (SGI).
- Basic Features:
  - C language interface
  - Graphics primitives: points, line segments, polygons. Defined by one or a group of vertices
  - State machine
  - Manipulation of frame buffer contents
- Other Features:
  - Clipping, shading, texture mapping, etc

Data Types

`GLbyte`	`signed char`
`GLshort`	`short`
`GLint`	`int`
`GLfloat`	`float`
…	

Naming

- Functions: **`gl`**`Foo()`
- Defined constant **`GL`**`_FOO`
- “Overloading”
  - `glColor3f()`
  - `glColor4f()`
  - `glColor3fv()`
  - …
- Functions from OpenGL Utility (GLU) Library: **`glu`**`Foo()`

Using Primitives

- `glColor*()`
- `glBegin(GLenum mode)`, `glEnd()`
  - `GL_POINTS`
  - `GL_POLYGON`
  - …
- `glVertex*()`
- `glFlush()`

Primitives Example Code

```
glColor3f(red, green, blue);

// draw a single-pixel dot to the
// window, at pixel coordinate (x,y)
glBegin(GL_POINTS);
  glVertex2i(x, y);
glEnd();

// draw an outlined triangle (many ways
// to do this) having vertices A, B,
// and C
glBegin(GL_LINE_STRIP);
  glVertex2i(Ax, Ay);
  glVertex2i(Bx, By);
  glVertex2i(Cx, Cy);
  glVertex2i(Ax, Ay);
glEnd();
```

Primitives Example Code (Cont’d)

```
// draw a filled triangle having
// vertices A, B, and C
glBegin(GL_POLYGON);
  glVertex2i(Ax, Ay);
  glVertex2i(Bx, By);
  glVertex2i(Cx, Cy);
glEnd();

glFlush(); // don't forget this!
```

Basic Transformations

- Matrix modes in OpenGL
  - Projection matrix
  - Modelview matrix

```
glMatrixMode(GL_MODELVIEW);
glPushMatrix();

// do the translation and rotation
...
// draw the brush strok
...

glPopMatrix();
```

Translation Example

```
glPushMatrix();
  glTranslate*(x, y, 0.0);
  // Note that the translation is
  // done outside glBegin and glEnd
  glBegin(...);
    glVertex2i(...);
    // ...
  glEnd(...);
glPopMatrix();
```

For rotation:

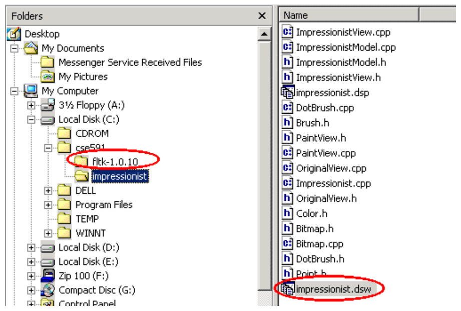
```
glRotate*(angle, axis.x, axis.y, axis.z);
```

Debugging

```
GLenum errCode = glGetError();
gluErrorString(errCode);
```

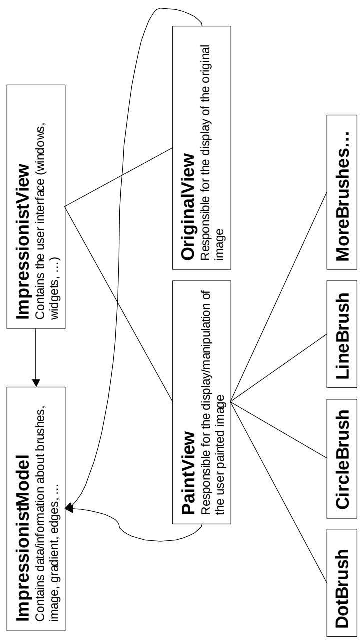
- A good place to insert the above lines is after a `glBegin`-`glEnd` block.

Impressionist



Folders
Desktop
My Documents
Messenger Service Received Files
My Pictures
My Computer
3½ Floppy (A:)
Local Disk (C:)
CDROM
cse591
fltk-1.0.10
impressionist
DELL
Program Files
TEMP
WINNT
Local Disk (D:)
Local Disk (E:)
Zip 100 (F:)
Compact Disc (G:)
Control Panel

Name
ImpressionistView.cpp
ImpressionistModel.cpp
ImpressionistModel.h
ImpressionistView.h
impressionist.dsp
DotBrush.cpp
Brush.h
PaintView.h
PaintView.cpp
OriginalView.cpp
Impressionist.cpp
OriginalView.h
Color.h
Bitmap.h
Bitmap.cpp
DotBrush.h
Point.h
impressionist.dsw


ImpressionistView
Contains the user interface (windows, widgets, ...)
ImpressionistModel
Contains data/information about brushes, image, gradient, edges, ...
OriginalView
Responsible for the display of the original image
PaintView
Responsible for the display/manipulation of the user painted image
DotBrush
CircleBrush
LineBrush
MoreBrushes...