Computer Graphics

- OpenGL-Programming -

Philipp Slusallek

OpenGL Preliminaries

Header Files

- #include <GL/gl.h>
- #include <GL/glu.h>
- #include <GL/glut.h>
  - Automatically includes gl.h, glu.h

Libraries

- -lopengl32 -lglu32 -lglut32

Enumerated Types

- OpenGL defines numerous types for compatibility
- GLfloat, GLint, GLenum, etc.

GLUT Basics

Application Structure

- **Configure and open window**
- **Initialize OpenGL state**
- **Register input callback functions**
  - render
  - resize
  - input: keyboard, mouse, etc.
- **Enter event processing loop**

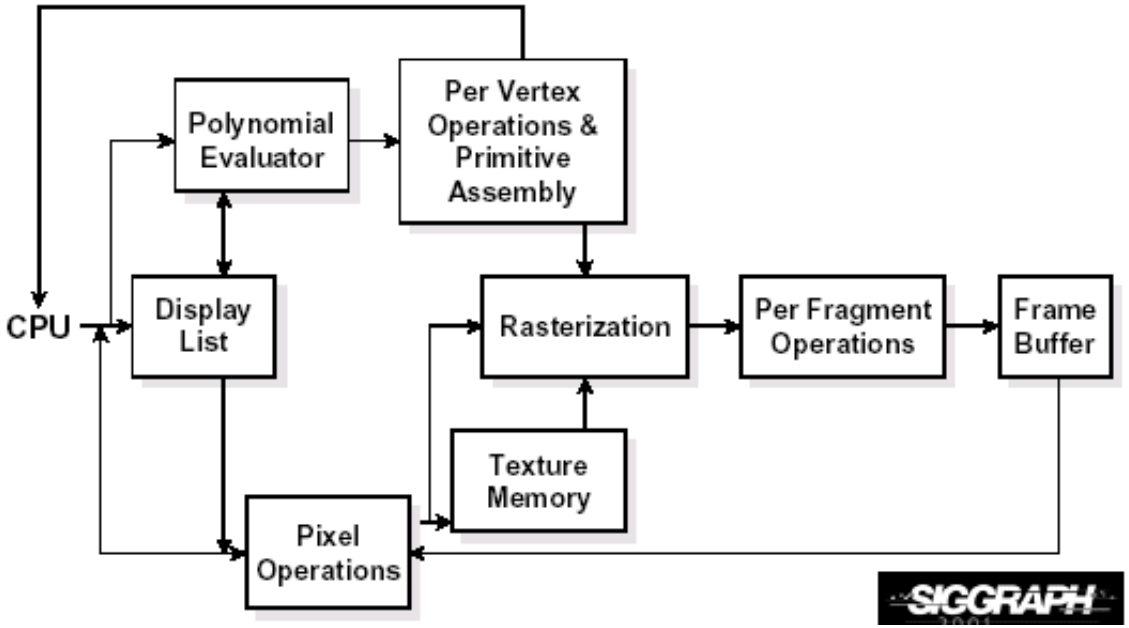
Main Template

```
int main(int argc, char** argv) {
    int mode = GLUT_RGB | GLUT_SINGLE;
    glutInit(&argc, argv);
    glutInitDisplayMode(mode);
    glutInitWindowSize(200, 200);
    glutInitWindowPosition(200, 200);
    glutCreateWindow("OpenGL Demo");
// …
    glutMainLoop();
    return 0;
}
```

OpenGL – State Machine

All rendering attributes are encapsulated in the OpenGL State

- rendering styles
- shading
- lighting
- texture mapping



Manipulating OpenGL State

Appearance is controlled by current state

```
for each ( primitive to render ) {
    update OpenGL state
    render primitive
}
```

Manipulating vertex attributes is most common way to manipulate state

glColor*() , glNormal*() , glTexCoord*(), …

Controlling the Current State

Setting State

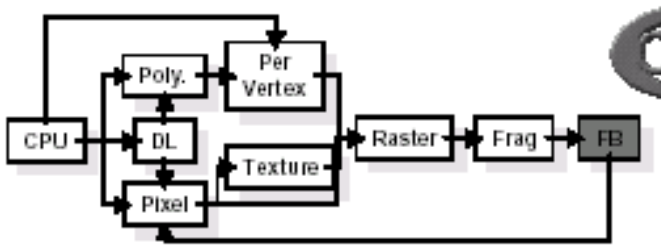
glPointSize(size);
glLineStipple(repeat, pattern);
glShadeModel(GL_ SMOOTH);

Enabling Features

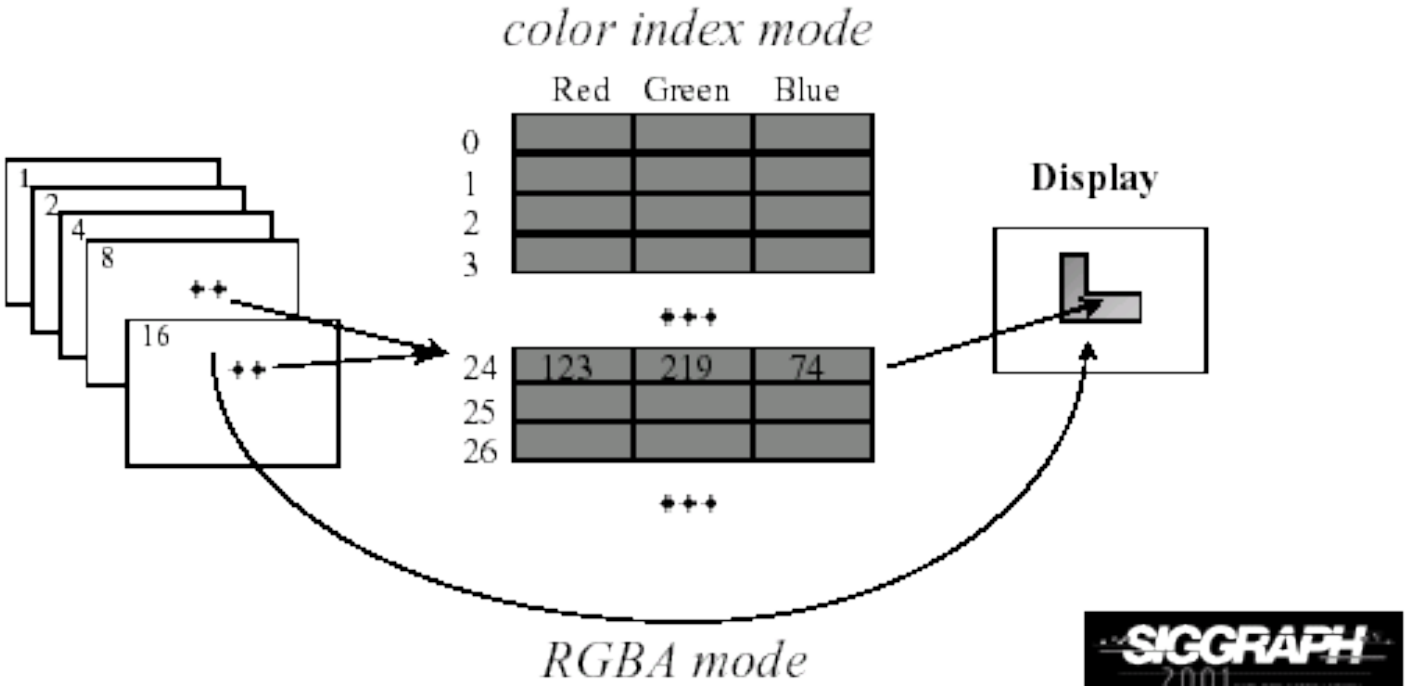
glEnable(GL_ LIGHTING);
glDisable(GL_TEXTURE_2D);

OpenGL Color Models

- **RGBA or Color Index**



  - glColor*() *or* glIndex*()
  - glutInitDisplayMode(GLUT_RGBA *or* GLUT_INDEX)



Initialization

Set up global state

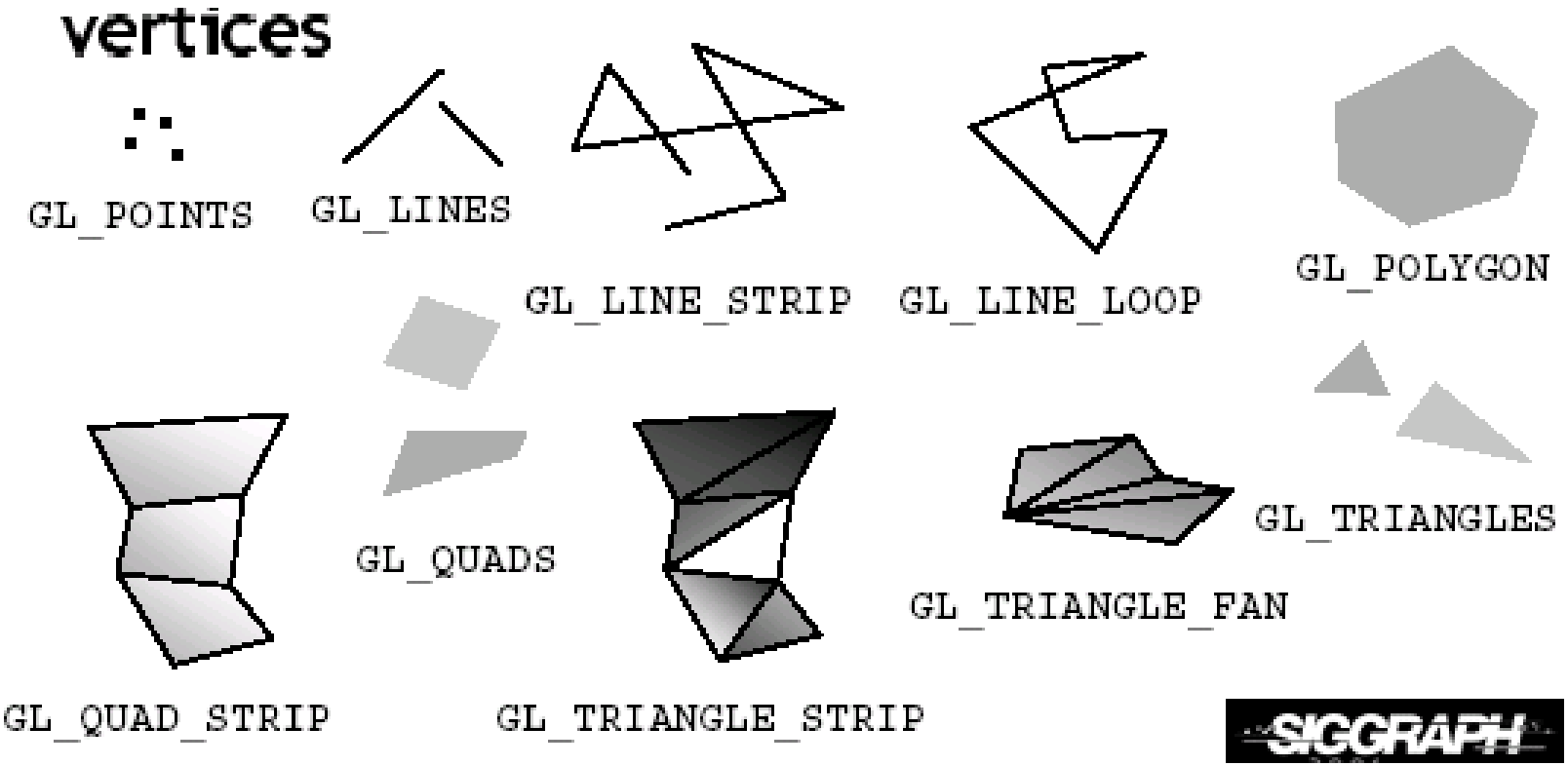
init();

- valid for entire execution time

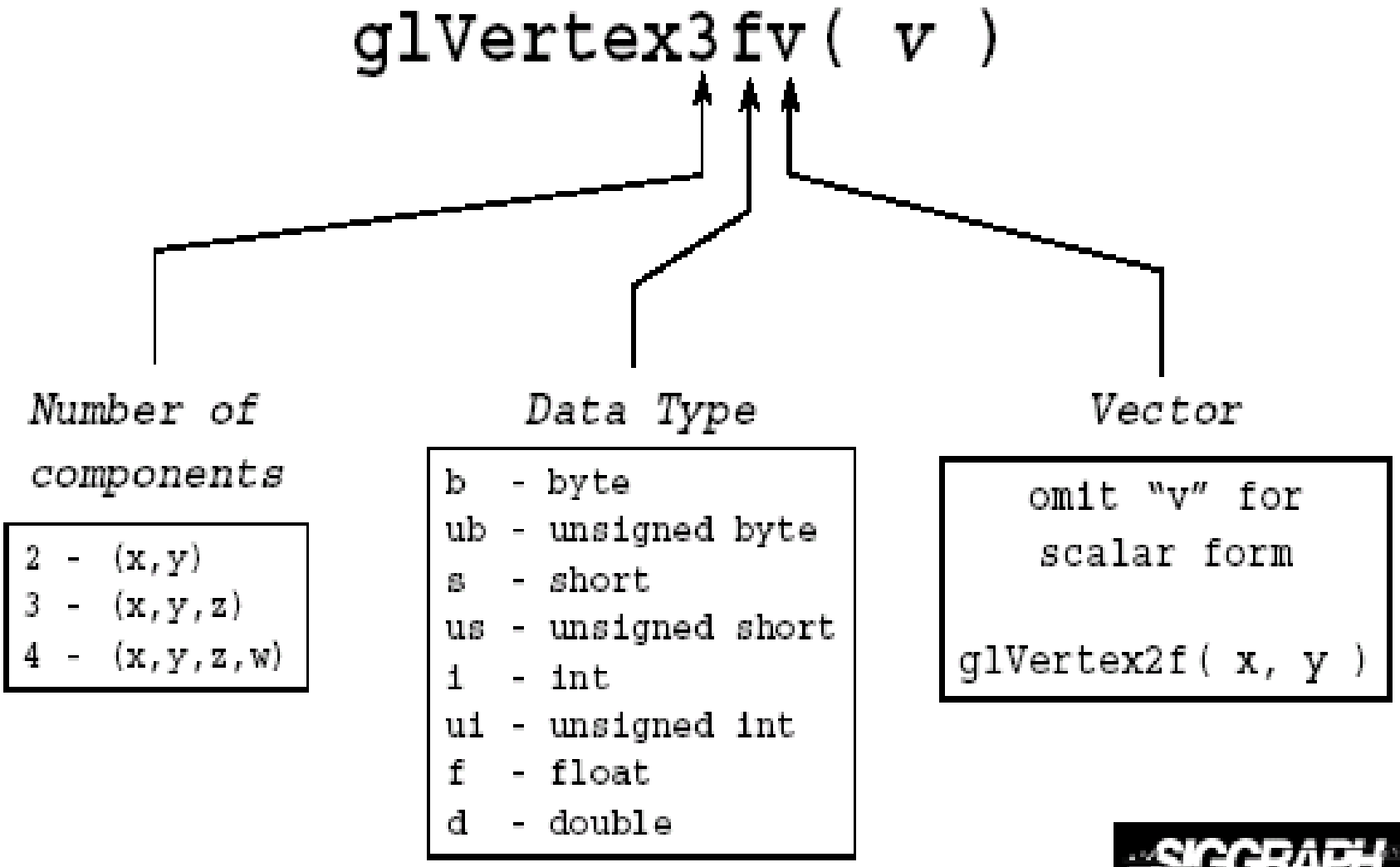
```
void init(void) {
    glClearColor(1.0, 1.0, 1.0, 1.0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(-1.0, 1.0, -1.0, 1.0, -1.0, 1.0);
}
```

Geometric Primitives

- All geometric primitives are specified by vertices



OpenGL Command Formats



Specifying Geometric Primitives

- **Primitives are specified using**

  glBegin(*primType*);
  glEnd();
  – *primType* determines how vertices are combined

```
GLfloat red, greed, blue;
Glfloat coords[3];
glBegin( primType );
for ( i = 0; i < nVerts; ++i ) {
  glColor3f( red, green, blue );
  glVertex3fv( coords );
}
glEnd();
```

OpenGL Primitive Types

- GL_POINTS
- GL_LINE_STRIP
- GL_LINES
- GL_LINE_LOOP
- GL_POLYGON
- GL_TRIANGLE_STRIP
- GL_TRIANGLES
- GL_TRIANGLE_FAN
- GL_QUADS
- GL_QUAD_STRIP

GLUT Callback Functions

Routine to call when something happens

- rendering
- user input
- animation
- window resize or redraw

“Register” callbacks with GLUT

```
glutDisplayFunc( display );
glutKeyboardFunc( keyboard );
glutIdleFunc( idle );
glutReshapeFunc( resize );
```

Rendering Callback

Do all of your drawing here

```
    glutDisplayFunc(display);

void display(void) {
    glClear(GL_COLOR_BUFFER_BIT);

    glBegin(GL_TRIANGLES);
        glColor3f(1, 0, 1); glVertex3f(-0.5, -0.5, 0.0);
        glColor3f(0, 0, 1); glVertex3f(-0.5, 0.5, 0.0);
        glColor3f(1, 0, 0); glVertex3f(0.5, 0, 0.0);
    glEnd();

    glFlush();
}
```

User Input Callback

React to key strokes

```
glutKeyboardFunc( keyboard );
```

```
void keyboard(unsigned char key, int x, int y) {
  switch (key) {
    case 27:
      exit(0); break;
    case '[':
      col = col < 0. ? 0. : col-0.1; glutPostRedisplay(); break;
    case ']':
      col = col > 1. ? 1. : col+0.1; glutPostRedisplay(); break;
    }
}
```

Global variable GLfloat col=0.;

In display() glColor3f(1, col, 0); glVertex3f(0.5, 0, 0.0);

Idle Callbacks

Use for animation and continuous update

```
glutIdleFunc( idle );
```

void idle(void) {

```
    t +=dt;
    glutPostRedisplay();
}
```

Global variables

```
GLfloat t = 0;
GLfloat dt= 0.001;
```

In display()

```
glColor3f( 0.5+0.5*cos(t), 0,1);
```

Callback Functions

- **glutDisplayFunc()**
  - called when pixels in the window need to be refreshed
- **glutReshapeFunc()**
  - called when the window changes size
- **glutKeyboardFunc()**
  - called when a key is struck on the keyboard
- **glutMouseFunc()**
  - called when the user presses a mouse button on the mouse
- **glutMotionFunc()**
  - called when the user moves the mouse while a mouse button is pressed
- **glutPassiveMouseFunc()**
  - called when the mouse is moved regardless of mouse button state
- **glutIdleFunc()**
  - called when nothing else is going on; very useful for animations