Exercise 1
Introduction to OpenGL

Alexandra Junghans

What we are going to do

- OpenGL
- Glut
- Small Example using OpenGl and Glut

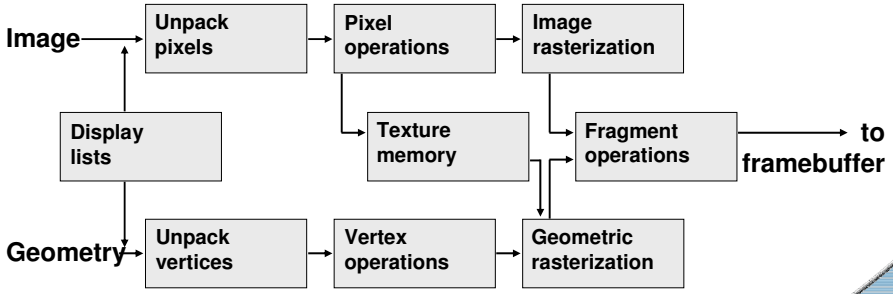
What is OpenGL?

- "most widely used and supported graphics API for portable and interactive 2D and 3D"
- Broad set of rendering, texture mapping and other powerful visualization functions
- Can render high-quality color images composed of geometric and image primitives
- Advantages:
  - window system and OS independent, stable, scalable, evolving, supported by many manufacturers
  - it's easy to use and well documented

OpenGL –Two Parts

- State Machine: Certain states control how OpenGL renders a scene
  - Colors, materials, light sources, etc.
- Specification of objects to be rendered:
  - Geometric primitives: Points, lines and polygons
  - Image primitives: Images and bitmaps
  - Separate pipeline for images & geometry ...linked through texture mapping

Data Flow



[fragment = pixel]

OpenGL initialization

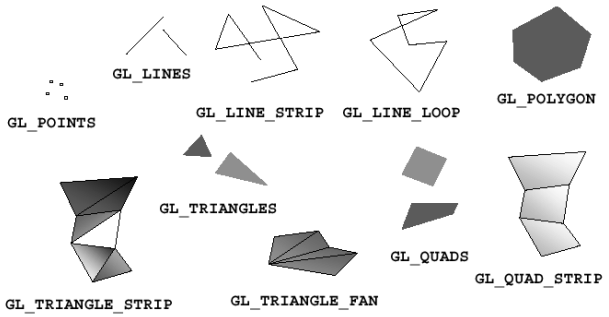
- Set up whatever state you are going to use

```
void init(void)
{
    glClearColor( 0.0, 0.0, 0.0, 1.0 );
    glClearDepth( 1.0 );

    glEnable( GL_LIGHT0 );
    glEnable( GL_LIGHTING );
    glEnable( GL_DEPTH_TEST );
}
```


OpenGL geometric primitives

- All geometric primitives are specified by vertices



Specifying geometric primitives

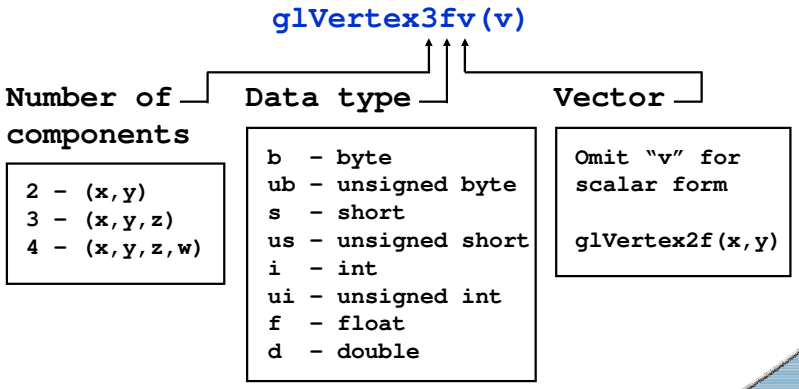
- Primitives are specified using:

```
glBegin(type);  ...  glEnd();
```

- Code example

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

OpenGL command formats



OpenGL Types

- OpenGL provides an own type system
  - Glfloat : 32-bit floating-point
  - Glint: a 32 bit integer
  - Glboolean: 8-bit unsigned integer

  And some more
- Why?
  - To ensure portability, therefore try to use them, although it´s a bit more typing work

More Information

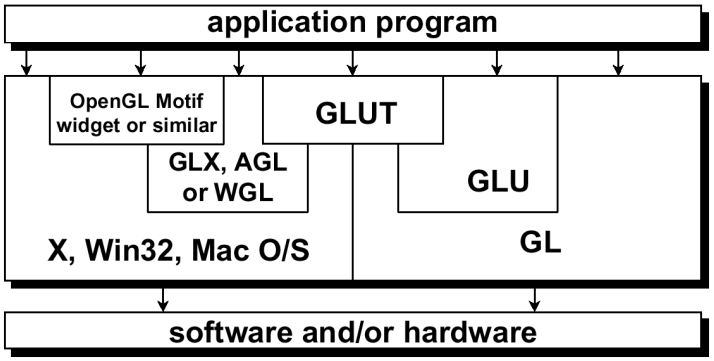
- OpenGl man pages!

  http://www.opengl.org/documentation/specs/man_pages/hardcopy/GL/html/

Related APIs

- GLU (OpenGL Utility Library)
  - Part of OpenGL
  - NURBS, tessellators, quadric shapes, etc.
- GLUT (OpenGL Utility Toolkit)
  - Portable windowing API
  - Many additional feature, e.g. glutWiredSphere()
  - not officially part of OpenGL
- AGL, GLX, WGL
  - Glue between OpenGL and windowing system

OpenGL and related APIs



GLUT callback functions

- Routine to call when something happens („event handler“)
  - Window resize or redraw
  - User input
  - Animation
- Must register callbacks with GLUT

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


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

Sample program

```
void main( int argc, char** argv )
{
    glutCreateWindow(argv[0]);
    glutInitDisplayMode(GLUT_RGB|GLUT_DOUBLE);
    init();

    glutDisplayFunc(display);
    glutReshapeFunc(resize);
    glutKeyboardFunc(key);
    glutIdleFunc(idle);

    glutMainLoop();
}
```

Rendering callback

- Do all of your drawing here

```
glutDisplayFunc(render);
```

- Code Example

```
void render(void)
{
    glClear( GL_COLOR_BUFFER_BIT );
    glBegin( GL_TRIANGLE_STRIP );
    glVertex3fv( v[0] ); glVertex3fv( v[1] );
    glVertex3fv( v[2] ); glVertex3fv( v[3] );
    glEnd();
    glutSwapBuffers();
}
```


Idle callback

- Use for animation and continuous update

```
glutIdleFunc(idle);
```

- Code example

```
void idle(void)
{
    t +=dt;
    glutPostRedisplay();
}
```


User input callback

- Process user input

```
glutKeyboardFunc(keyboard);
```

- Code example

```
void keyboard(char key, int x, int y)
{
    switch(key) {
    case q: case Q:
          exit(); break;
    case r: case R:
          rotate = GL_TRUE; break;
    }
}
```


Double Buffering

- The drawing process is visible when using only one buffer
- Nasty for animation where we want a certain frame rate without flickering
- Therefore most graphic cards have two or more frame buffers → display the front buffer until the back buffer is drawn completely, then swap the buffers

Useful links

- www.opengl.org
- nehe.gamedev.net
- www.naturewizard.com
- http://en.wikipedia.org/wiki/OpenGL