



A Short Intro to OpenGL on Darwin

September 2008
CocoaHeads Hong Kong



Agenda

- Overview of OpenGL
- Basic Concepts
- How to get a triangle
- How to get a texture



OpenGL



- it's a rasterizer
- deals mostly in triangles
- well supported by (all) hardware
- => it's lowlevel, and fast!



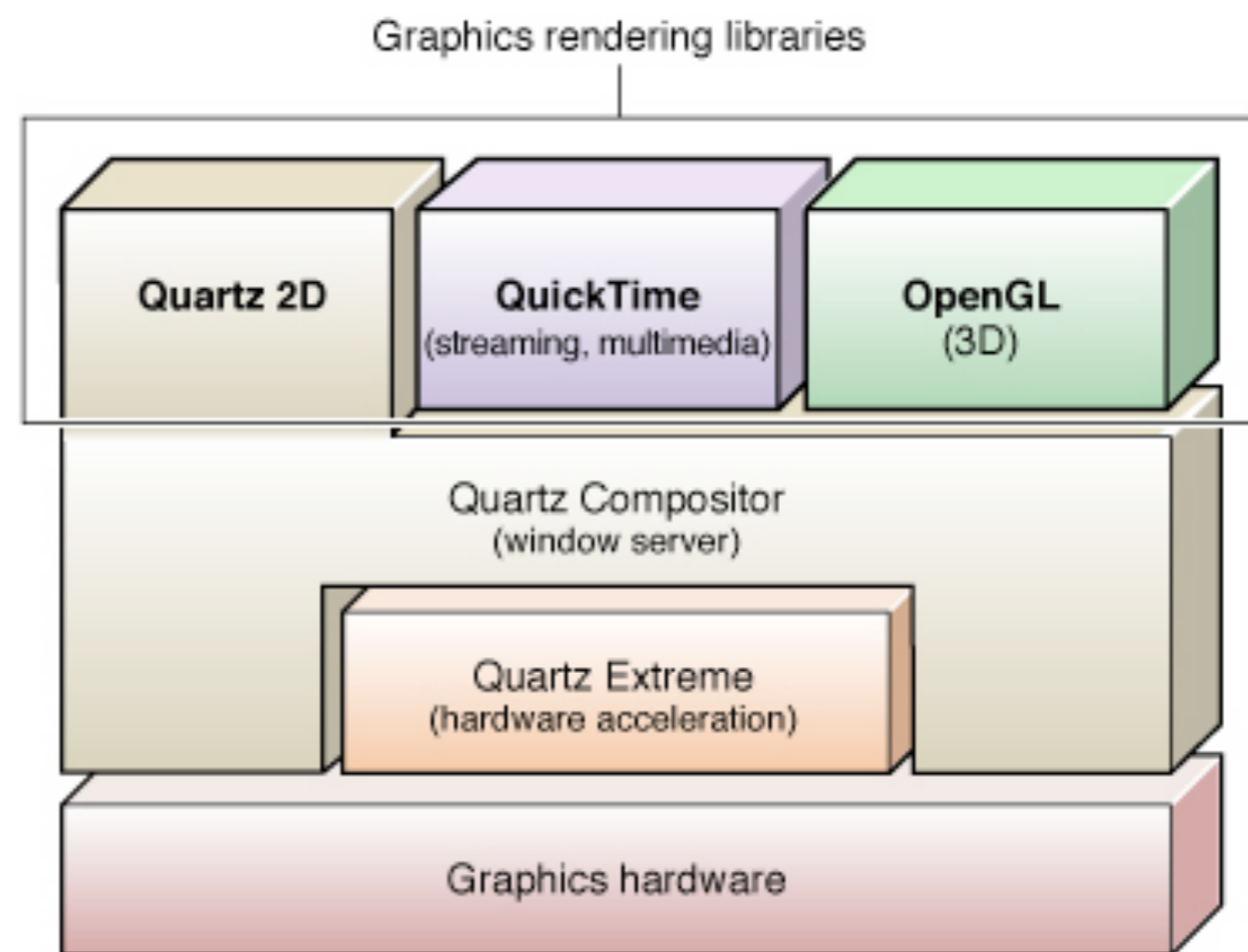
Related Technologies

- Quartz2D, for two dimensional stuff
- CoreAnimation, for highlevel effects
- OpenGL ES, for embedded platforms
- Direct3D, on some platforms



OpenGL on Mac OS X

- Quartz Extreme is implemented in OpenGL
- Multiple OpenGL Contexts in OS X at any given time
- it's all hidden from you, just subclass `NSOpenGLView`





OpenGL in Cocoa

1. create a Cocoa Application project
2. add OpenGL.framework
3. make the view extend NSOpenGLView
4. change class in Interface Builder
5. put OpenGL drawing code into
 - (void) drawRect:(NSRect) rect {}





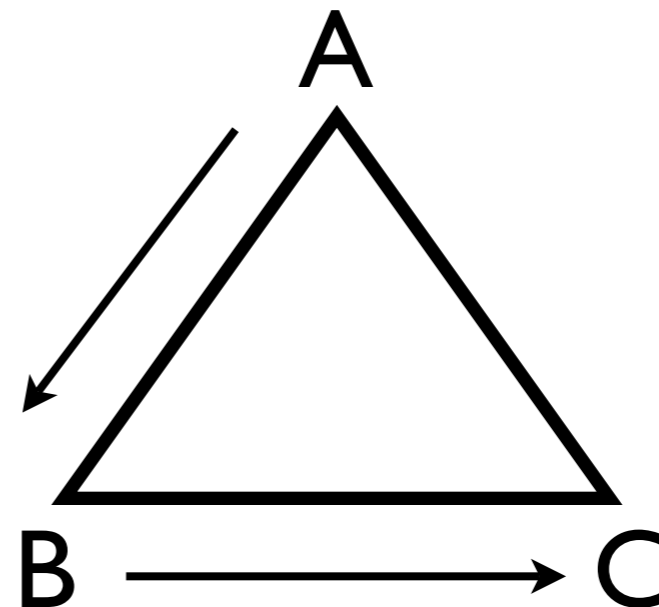
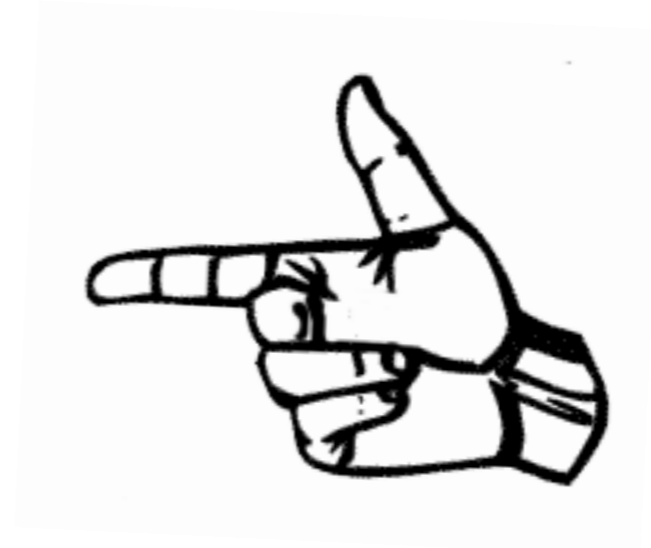
Rendering a Frame

1. clear buffers
2. specify geometry and transformations
3. flush buffer
4. goto 1. (for the next frame, that is)



Conventions

- right hand rule applies
- vertices are given counterclockwise





Specifying Geometry

1. Set up data in memory

```
float v[] = { 1.0f, 1.0f, ... };
```

2. Specify data to be used for vertices, normals, colors

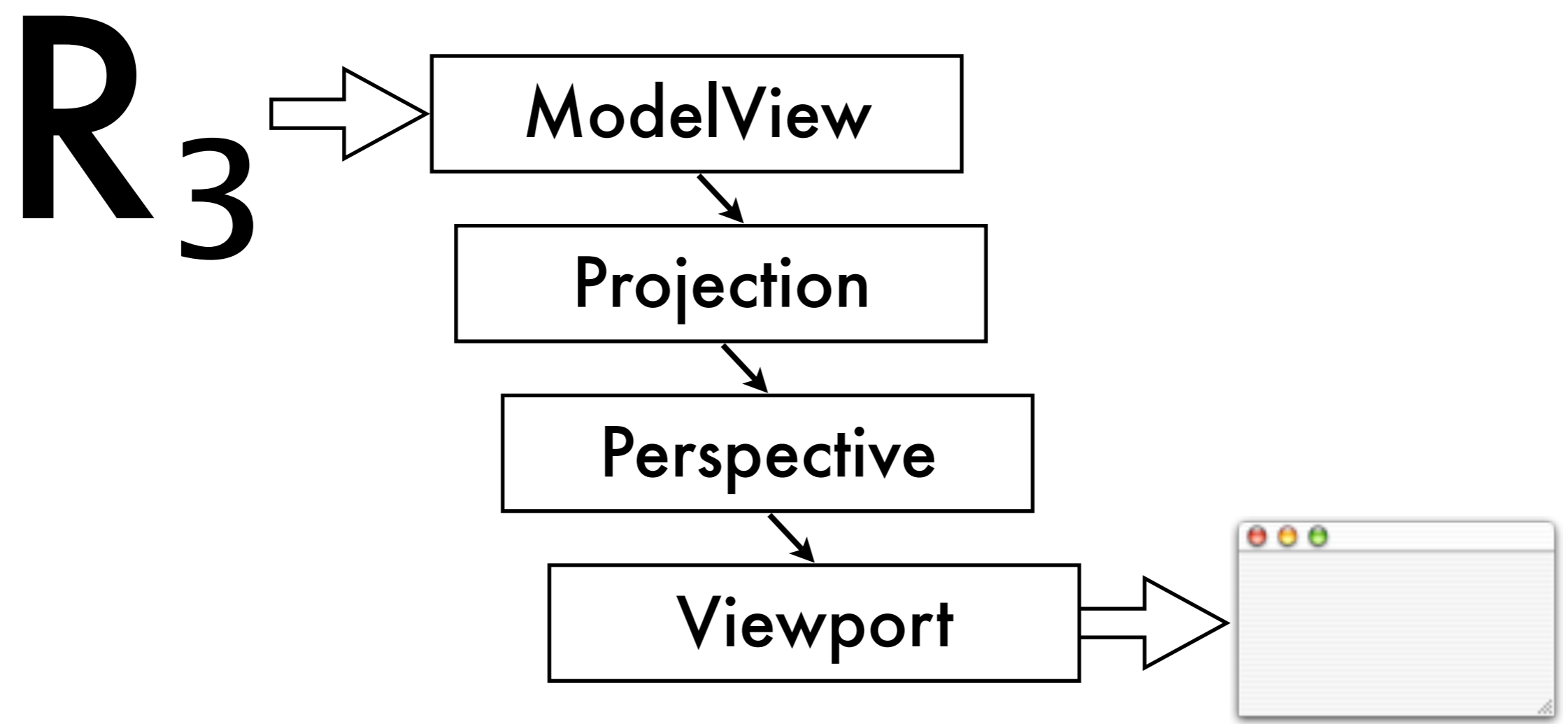
```
glVertexPointer(3, GL_FLOAT, 0, v);
```

3. Draw the data

```
glDrawArrays(GL_TRIANGLES, 0, 3);
```



Transformation Sequence





Matrix Operations

- `glTranslate(x, y, z);`
- `glRotatef(α , x, y, z);`
- `glScale(x, y, z);`

- `glPushMatrix(); glPopMatrix();`



Textures

- OpenGL knows only buffers
 - => need to load / create the image data ourselves
- Fortunately, there's `NSBitmapImageRep`



Setting Up Textures

1. load data from file

```
UIImage -initWithContentsOfFile;
```

2. create texture name, bind texture

```
glGenTextures(); glBindTexture();
```

3. load into GL texture buffer

```
glTexImage2D();
```

```
NSBitmapImageRep -bitmapData;
```



Drawing Textures

1. Specify Texture Coordinates

```
glTextureVertexPointer();
```

2. Bind Texture

```
glBindTexture();
```

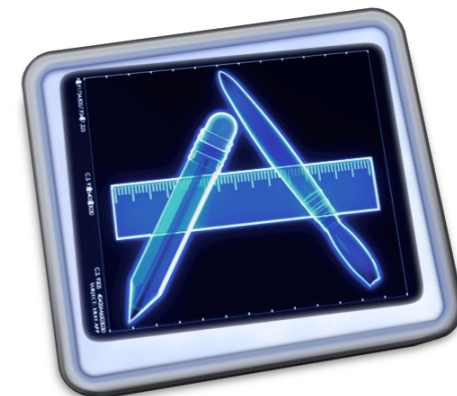
3. draw triangles as before

```
glDrawArrays();
```



Useful Tools

- `glGetError();`
`gluErrorString();`
- OpenGL Profiler
- Sampler Instrument
- OpenGL Driver Instrument





(My) Best Practices

- use C++, for operator overloading
- use a vector class, define addition, multiplication operators etc
- use quaternions for rotation

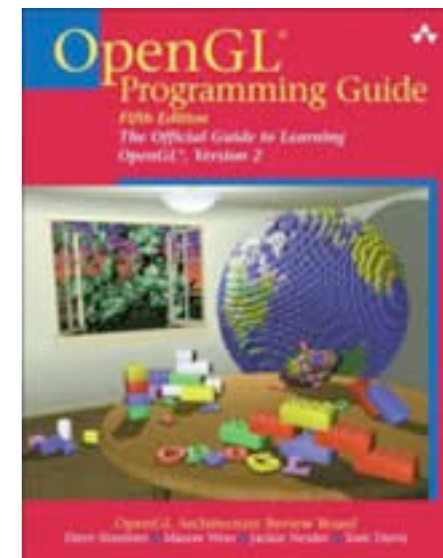


References

- `man glTranslate` etc
- OpenGL Programming Guide for Mac OS X (Apple Developer Connection)
- OpenGL Programming Guide

Shreiner et al, Addison-Wesley, ISBN 978-0321481009

The 'Red Book'





Questions?

?

?

?

?

?