OpenGL and Shaders
Advanced Graphics and Interaction
OpenGL and Shaders

- OpenGL's Fixed Function Pipeline provides limited functionality
- With shaders, you have complete control over the results
OpenGL and Shaders

Loading Textures

- Loading textures in Cg is the same as in plain OpenGL
- Using them is not!

The OpenGL way:
```
glBindTexture(GL_TEXTURE_2D, texture);
```

The Cg way:
```
cgGLSetTextureParameter(cgTexture, texture);
```
OpenGL and Shaders
Loading textures - Multitexture

The OpenGL way:

```c
#define __ARB_ENABLE true
...
glActiveTextureARB(GL_TEXTURE0_ARB);
glBindTexture(GL_TEXTURE_2D, texture1);
glTexEnvi(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_REPLACE);
glActiveTextureARB(GL_TEXTURE1_ARB);
glBindTexture(GL_TEXTURE_2D, texture2);
glTexEnvi(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_COMBINE);

glTexEnvi(GL_TEXTURE_ENV, GL_COMBINE_RGB, GL_MODULATE);
glTexEnvi(GL_TEXTURE_ENV, GL_SOURCE0_RGB, GL_PREVIOUS);
glTexEnvi(GL_TEXTURE_ENV, GL_SOURCE1_RGB, GL_TEXTURE);
glTexEnvi(GL_TEXTURE_ENV, GL_OPERAND0_RGB, GL_SRC_COLOR);
glTexEnvi(GL_TEXTURE_ENV, GL_OPERAND1_RGB, GL_SRC_COLOR);
```
OpenGL and Shaders
Loading textures - Multitexture

The Cg Way:

cgGLSetTextureParameter(cgTexture0, texture1);
cgGLSetTextureParameter(cgTexture1, texture2);

• Combining (blending) the textures is done in the fragment shader
• How this is done is entirely up to you!
OpenGL and Shaders

OpenGL - Pros
● Works on older hardware
● Easier to debug

OpenGL - Cons
● Limited functionality; not flexible
● Some functionality requires extensions
● Does less with more code

Shaders - Pros
● Powerful
● Gives more control; very flexible
● Does more with less code

Shaders - Cons
● Requires newer hardware
● Harder to debug
● Harder to work with
OpenGL and Shaders

Environment Mapping

- Cheap way of reflecting the environment in real-time
- Uses special textures to represent the environment mapped onto simple shapes
- Unable to reflect other objects
- Only an approximation!
OpenGL and Shaders

Environment Mapping

Sphere mapping

- Texture represents the environment as reflected on a sphere
- Simple, but prone to artefacts due to distortion

\[
\begin{align*}
    r &= v - 2n(n \cdot v) \\
    m &= \sqrt{r_x^2 + r_y^2 + (r_z + 1.0)^2} \\
    s &= \frac{r_x}{2m} + 0.5 \\
    t &= \frac{r_y}{2m} + 0.5
\end{align*}
\]
OpenGL and Shaders

Environment Mapping

Cube mapping
- Texture represents the environment as reflected on a cube
- Texture consists of six parts
- Less artifacts