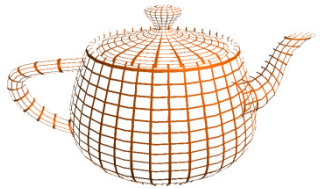
sim

Aula Teórico-Prática

GLSL - Texturas

António Ramires Fernandes - CG



GLSL - Texturas

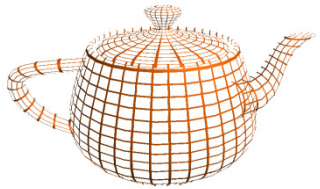
- Aplicação OpenGL envia coordenadas de textura através dos atributos:

```
attribute vec4 gl_MultiTexCoord0;
...
attribute vec4 gl_MultiTexCoord7;

uniform mat4 gl_TextureMatrix[gl_MaxTextureCoords];
```

- No vertex shader é necessário passar a coordenada de textura para o fragment shader:

```
gl_TexCoord[0] = gl_MultiTexCoord0;
```



GLSL - Texturas

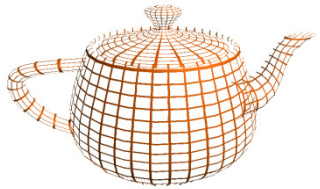
- Vertex Shader

```
void main() {
    gl_TexCoord[0] = gl_MultiTexCoord0;
    gl_Position = ftransform();
}
```

- Fragment Shader

```
uniform sampler2D tex;

void main() {
    vec4 color = texture2D(tex,gl_TexCoord[0].st);
    gl_FragColor = color;
}
```



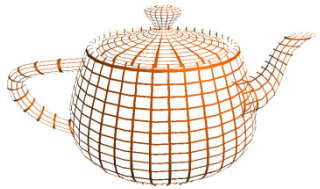
GLSL - Texturas

- Pretende-se agora implementar um par de shaders que permita afectar a cor da textura com a iluminação:

- Considerando o seguinte vertex shader escreva o seu par, ou seja o fragment shader

```
varying vec3 lightDir,normal;

void main() {
    normal = normalize(gl_NormalMatrix * gl_Normal);
    lightDir = normalize(vec3(gl_LightSource[0].position));
    gl_TexCoord[0] = gl_MultiTexCoord0;
    gl_Position = ftransform();
}
```



GLSL - Texturas

- Assuma agora que se pretende combinar duas texturas, presentes nas unidades 0 e 1.

Texture Unit 0

Texture Unit 1

Textured Cube

- Pretende-se que a segunda unidade de textura tenha sempre a mesma intensidade, independentemente da iluminação