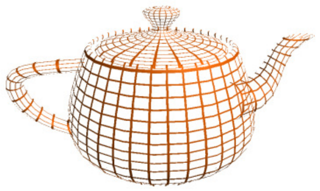
Aula Teórico-Prática

GLSL

António Ramires Fernandes - Computação Gráfica 07/08



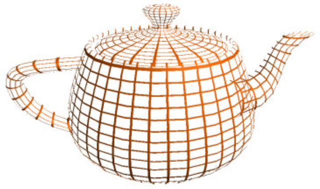
GLSL

- Shader Designer (www.typhoonlabs.com)
  - New Project
  - Vertex Shader

```
void main()
{
   gl_Position = ftransform();
}
```

  - Fragment Shader

```
void main()
{
   gl_FragColor = vec4(1.0,1.0,1.0,1.0);
}
```



GLSL

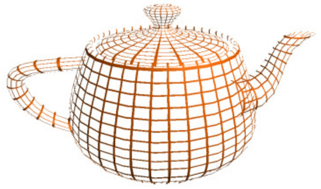
- Escreva o seguinte par de shaders e analise o resultado:

- Vertex Shader

```
void main()
{
   if (gl_Vertex.x > 0.0 && gl_Vertex.y > 0.0)
           gl_FrontColor = vec4(1.0,0.0,0.0,1.0);
   else if (gl_Vertex.x < 0.0 && gl_Vertex.y < 0.0)
           gl_FrontColor = vec4(1.0,0.0,0.0,1.0);
   else
           gl_FrontColor = vec4(0.0,1.0,0.0,1.0);
   gl_Position = ftransform();
}
```

- Fragment Shader

```
void main() {
     gl_FragColor = gl_Color;
}
```



GLSL

- Considere agora os seguintes shaders. Compare o resultado com o exemplo do slide anterior.
- Vertex Shader

```
varying vec4 vertexCamera;

void main()
{
   vertexCamera = gl_Vertex;
   gl_Position = ftransform();
}
```

- Fragment Shader

```
varying vec4 vertexCamera;

void main()
{
   vec4 color;
   if (vertexCamera.x > 0.0 && vertexCamera.y > 0.0)
            color = vec4(1.0,0.0,0.0,1.0);
   else if (vertexCamera.x < 0.0 && vertexCamera.y < 0.0)
            color = vec4(1.0,0.0,0.0,1.0);
   else
            color = vec4(0.0,1.0,0.0,1.0);

   gl_FragColor = color;
}
```

GLSL

- Assuma que a intensidade da luz reflectida por um objecto é proporcional ao produto interno entre a direcção da luz e a normal da superfície.

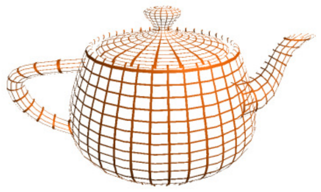
```
varying vec4 vertexCamera;
varying float intensidade;

void main()
{
            vec3 normal = normalize(gl_NormalMatrix *
gl_Normal);
            intensidade = max(dot(vec3(0,0,1),normal),0.0);
            vertexCamera = gl_Vertex;
            gl_Position = ftransform();
}
```

```
varying vec4 vertexCamera;
varying float intensidade;

void main()
{
            vec4 color;
            if (vertexCamera.x > 0.0 && vertexCamera.y > 0.0)
                        color = vec4(1.0,0.0,0.0,1.0);
            else if (vertexCamera.x < 0.0 && vertexCamera.y < 0.0)
                        color = vec4(1.0,0.0,0.0,1.0);
            else
                        color = vec4(0.0,1.0,0.0,1.0);

            gl_FragColor = color * intensidade;
}
```



Exercício

- Considerando o exemplo do acetato anterior, escreva um shader em que os cálculos da intensidade sejam realizados no fragment shader.