

Atributs (enviats pel viewer; per usar-los cal declarar-los als shaders):

layout (location = 0) in vec3 vertex; // similar a gl_Vertex (però 3D)

layout (location = 1) in vec3 normal; // idèntic a gl_Normal

layout (location = 2) in vec3 color; // similar a gl_Color (però RGB)

layout (location = 3) in vec2 texCoord; // similar a gl_MultiTexCoord0

Uniforms (enviats automaticament pel viewer; per usar-los cal declarar-los als shaders):

uniform mat4 modelMatrix;

uniform mat4 viewMatrix;

uniform mat4 projectionMatrix;

uniform mat4 modelViewMatrix;

uniform mat4 modelViewProjectionMatrix;

uniform mat4 modelMatrixInverse;

uniform mat4 viewMatrixInverse;

uniform mat4 projectionMatrixInverse;

uniform mat4 modelViewMatrixInverse;

uniform mat4 modelViewProjectionMatrixInverse;

uniform mat3 normalMatrix;

uniform vec4 lightAmbient; // similar a gl_LightSource[0].ambient

uniform vec4 lightDiffuse; // similar a gl_LightSource[0].diffuse

uniform vec4 lightSpecular; // similar a gl_LightSource[0].specular

uniform vec4 lightPosition; // similar a gl_LightSource[0].position

// (sempre estarà en eye space)

uniform vec4 matAmbient; // similar a gl_FrontMaterial.ambient

uniform vec4 matDiffuse; // similar a gl_FrontMaterial.diffuse

uniform vec4 matSpecular; // similar a gl_FrontMaterial.specular

uniform float matShininess; // similar a gl_FrontMaterial.shininess

uniform vec3 boundingBoxMin; // cantonada mínima de la capsa englobant

uniform vec3 boundingBoxMax; // cantonada màxima de la capsa englobant

uniform vec2 mousePosition; // coordenades del cursor (window space)

// origen a la cantonada inferior Esquerra

Variables “varying”: el VS les envia al FS

- Son de sortida al VS
- Son d'entrada al FS
- Exemples: color, normal, coord del vertex, coord de textura..

Les tasques del VS:

- Transformar el vertex (object space -> clip space)
- Transformar i normalitzar la normal
- Calcular la il·luminació del vèrtex
- Generar o passar les coordenades de textura pel vèrtex.

El fragment shader s'executa per cada fragment que produeix cada primitiva.

Les tasques del FS:

- Accedir a textura
- Incorporar el color de la textura
- Incorporar efectes a nivell de fragment

Càlculs d'il·luminació:

Phong -> $K_e + K_a(M_a + I_a) + K_d I_d(N * L) + K_s I_s(R * V)^S$

Blinn-Phong -> $K_e + K_a(M_a + I_a) + K_d I_d(N * L) + K_s I_s(N * H)^S$

On K = material, I = llum.

Quan normalitzar:

- Els vectors (N, L, R, V, H) han de ser unitaris.
- La longitud d'un vector no es preserva
- On normalitzar? Immediatament abans de fer els càlculs que assumeixen que el vector es unitari.

Tema textures:

Al FS

Un sampler 2D és una textura 2D (uniform sampler2D myMap)

Texture -> retorna un vec4 amb el color RGBA

Gl_FragColor = texture(myMap, vtexCoord);

Uniform float time; // temps des de la darrera compilació.

Fract(x) retorna la part fraccionaria de x calculada com $x - \text{floor}(x)$

Mod(x,y) calculat com $x - y * \text{floor}(x/y)$

Matrius mat3 m = mat3(vec3, vec3, vec3)

Interpol·lacio pel fragment

Fract(gl_FragCoord.x) = 0.5 (per la y igual)

Float fx = dFdx(...) la derivada

Float step(float edge, float x)

Float smoothstep(float edge0, float edge1, float x)