Gràfics

Informe
Pràctica 1

Oleksandr Danylenko
Xavi Cano
A05

Index

1.Material

1.1 - Comentaris:

Respecte a la part del material:

- Hem fet la part opcional de llegir el material per fitxer, amb la diferencia de que al acavar de llegir el fitxer .obj comprovem si existeix un fitxer .mtl amb el mateix nom. En cas d'existir llegim les components del fitxer del material agafant Ns com Shininess, Ka, kd, Ks. La resta de dades del .mtl no les llegim.

(objecte.cpp , funció readMtl, Linea 272)

- Si no existeix un fitxer de material el creem amb els seguents valors per defecte:

```
this->Ambient =  vec3 (0.2, 0.2, 0.2);
this->Diffuse =  vec3 (0.8, 0.0, 0.0);
this->Specular  = vec3 (1.0, 1.0, 1.0);
this->Shininess = 200;
this->Alpha  = 1.0;
```

(objecte.cpp , funció readMtl(), Linea 327)

2.Llums

2.1 - Comentaris:

Respecte a la part de Llums:

- En glwidget.cpp creem les llums (una de cada tipus) i se les asignem al mon.

(glwidget.cpp , funció initializeGL(), Linea 327)

- Totes les llums tenen per defecte els seguents valors, es troben en estat ON:

```
this->AmbientIntensity = vec4 (0.2,0.2,0.2,1.0);
this->SpecularIntensity = vec4 (1.0,1.0,1.0,1.0);
this->DiffuseIntensity = vec3 (0.8,0.8,0.8);
this->Position = vec4 (2.0,2.0,2.0,1.0);
this->Direction = vec4 (0.0,0.0,10.0,1.0);
this->Angle = 0.0;
this->Alpha = 1.0;
this->a = 0.0;
this->b = 0.0;
this->c = 1.0;
```

(llum.cpp , constructor de llum, Linea 104)

- En cas de que les tres llums es trobin en estat OFF es mostra la silueta, ja que tenim implementada l'ambient global(ambientGlobal).

3.Shaders

3.1 - Comentaris:

Respecte a la part de Shaders:

- Gouraud Shading:

En aquest shading implementem les formules explicades a teoria i el model de Bling Phong en el vShader.

- Phong Shading:

En aquest shading implementem les formules explicades a teoria i el model de Bling Phong en el fShader.

- Toon Shading:

En aquest shading en lloc d'utilitzar la direcció, utilitzem la posició en el calcul de l'intensitat, d'aquesta forma al cambiar la posició de la llum es canvia l'efecte del toon shading.
La forma correcta de fer-ho amb la direcció la tenim comentada al costat, però per fer l'entrega de la pràctica hem deixat la posició ja que ens va be per tal de realitzar proves amb els slides. Això es calcula a nivell de fShader.
En aquest shader només utilitzem una llum per al calcul, en el nostre cas la 2(sent 0 la primera) per poder fer les proves més facilment.
Tampoc hem implementat textura perquè no li trobem utilitat però si que sería necessari posar-la, ho fariem com sempre, pasant-la amb uniform i fent el link abans del shader.

4.Textures

4.1 - Comentaris:

Respecte a la part de Textures:

- Els calculs de (u,v) de les textures els hem fet a nivell de CPU , concretament en el make de l'objecte ja que no el rotem. En cas d'implementar la rotació ho hauriem d'implementar a nivell de GPU, o sigui al vShader.

- Gouraud Shading amb textura:

Esta implementat apartir del mateix shading de Gouraud amb la diferencia de que la textura s'aplica en cas de que la variable "conTextura" sigui true, tota la resta es lo mateix excepte la comprovació d'aquesta variable, que això es fa al fShader amb un if i dins d'aquest es fa la ponderació de textures i color.

*gl_FragColor = 0.25*color + 0.75*texture2D(texMap, v_texcoord);*

- Phong Shading amb textura:

Esta implementat apartir del mateix shading de Phong amb la diferencia de que la textura s'aplica en cas de que la variable "conTextura" sigui true, tota la resta es lo mateix excepte la comprovació d'aquesta variable, que això es fa al fShader amb un if i dins d'aquest es fa la ponderació de textures i color agafant la component difusa del material.

*diffuseMat = diffuseMat * 0.2 + 0.8 *texture2D(texMap, v_texcoord).rgb*

5.Comentaris generals

- Per tal de poder fer els càlculs de (u,v) hem fet un include del valor de PI.

- No hem pogut fer un array de programs i inicialitzar tots els shaders al principi degut a problemes d'inserció d'un program en un array , per això utilitzem InitShader cada vegada que activem un shader. Suposem que si un shader esta compilat una cop ja no es tornarà a compilar més.

- En l'apartat de càlcul de normals, el que fem es recorrer totes les cares, calculem les seves normals i el resultat el sumem en un vector de points de normals que tenim. Tenim un vector de normals on guardem la suma dels vertexs.
També tenim un vector de id's de vèrtexs i després de fer la suma fem la mitjana.
Just abaix esta comentat el flat shading , on es calcula la normal per a cada cara i no per a cada vèrtex.

6.Funcionalitats per implementar

- Bump-mapping:

Ens falta acabar de fer-ho, voliem fer-ho utilitzant un doble mapping esferic, però degut a que la clase de teoria va ser dilluns i l'entrega es dimarts no ens ha donat temps.

- Rotacio de l'esfera:

També ens falta fer la rotació de l'esfera mitjançant les fletxes del teclat, voliem fer-ho utilitzant una matriu de rotació que s'aplicaría a tots els punts a través de metodes com RotateY, que ja esta implementat en el mat.h.