Examen Parcial de Gràfics Curs 2015-16 Q2

Nom i Cognoms:

Exercici 1

Aquesta funció implementa el típic model d'il·luminació (Lambert+Phong):

```
vec4 light(vec3 N, vec3 V, vec3 L)
{
    V=normalize(V);
    L=normalize(L);
    vec3 R = normalize( 2.0*dot(N,L)*N-L );
    float NdotL = max( 0.0, dot( N,L ) );
    float RdotV = max( 0.0, dot( R,V ) );
    float Idiff = NdotL;
    float Ispec = 0;
    if (NdotL>0) Ispec=pow( RdotV, matShininess );
    return
        matAmbient * lightAmbient +
        matDiffuse * lightDiffuse * Idiff +
        matSpecular * lightSpecular * Ispec;
}
```

Descriu clarament la interpretació geomètrica dels paràmetres V i L de la funció.

= vector del punt cap a l'observador

L = vector del punt cap a la font de llum

Exercici 2

Indica quins tres vectors hauria de passar el VS al FS per que aquest últim pugui passar un vector qualsevol de tangent space a object space.

T, B, N (tangent, bitangent, normal).

Exercici 3

Sigui P(u,v) la representació paramètrica d'una superfície. Sigui F(u,v) un mapa d'elevacions. Escriu l'expressió que permet calcular la superfície P'(u,v) que resulta de pertorbar P segons el mapa d'elevacions, tal i com es fa servir en *displacement mapping*.

P'(u,v) = PP (u,v) + F(u,vv)*N(u,v) /||| N(u,v)||

Exercici 4

Aquesta equació relaciona, per bump mapping, la normal de la superfície pertorbada amb la normal original:

$$\mathbf{N}' = \frac{\partial \mathbf{P}}{\partial u} \times \frac{\partial \mathbf{P}}{\partial v} + \frac{\frac{\partial F}{\partial u}\left(\mathbf{N} \times \frac{\partial \mathbf{P}}{\partial v}\right)}{|\mathbf{N}|} + \frac{\frac{\partial F}{\partial v}\left(\frac{\partial \mathbf{P}}{\partial u} \times \mathbf{N}\right)}{|\mathbf{N}|} + \frac{\frac{\partial F}{\partial u}\frac{\partial F}{\partial v}(\mathbf{N} \times \mathbf{N})}{|\mathbf{N}|^2}$$

(a) Indica, dels quatre termes, quin avalua a un vector nul. Justifica la resposta.

El darrrer terme, j a que NxN=**00**

(b) Encercla a l'equació anterior un parell de vectors que siguin tangents a la superfície original.

Els veectors de la fforma Nx... óó ...xN

Exercici 5

Tenim un bump map on cada texel conté les dues components (du, dv) del gradient del camp d'elevacions F(u,v). Indica clarament (ex. codi GLSL) com podem calcular les components de la normal pertorbada N', en *espai tangent,* a partir de (du,dv).

vec3 N' = normalize(-du, -dv, 1)

Exercici 6

Aquí teniu una llista d'etapes/tasques del pipeline gràfic, ordenades per ordre alfabètic. Torna-les a escriure a la dreta, però ordenades segons l'ordre al pipeline gràfic:

- Crida a dFdx
- Declaració dels atributs vèrtex, normal, texCoord...
- Depth test
- Rasterització

Declaracció dels atri bbuts (VS)
Rasteritzzació
Crida a ddFdx (FS)
Depth tee st

Exercici 7

Indica i declara en GLSL 3.30 core els dos vec3 cal que el VS passi al FS per tal d'usar el model de Phong al FS.

Al VS:

out vec3 vnormal;

out vec3 pos; (vec4 també és correcte))

Al FS:

in vec3 vnormal;

in vec3 pos; (vec4 també és correcte))

Exercici 8

Completa aquest FS de forma que, amb la textura de l'esquerra, produeixi la imatge de la dreta, quan s'aplica a l'objecte plane.obj, que té coordenades de textura del (0,0) al (1,1).

```
void main() {
        vec2 f = vec2(....................., .....................);
        fragColor = texture(colormap, f * vtexcoord);
}
```

vec2((-3, 1)

Exercici 9

Aquest fragment de codi està generant coordenades de textura a partir de dos plans S, T: vec2

```
st = vec2(dot(vec4(vertex,1.0), planeS), dot(vec4(vertex,1.0), planeT));
```

Si tenim aquests uniforms,

```
uniform vec4 planeS = vec4( 4, 0, 0,
0 ); uniform vec4 planeT = vec4( 0, 1,
0, 0 );
```

Quines coordenades de textura (s,t) es calcularan pel vèrtex (2, 12, 127, 1)?

-> (4*2, 1*12) = (8, 12)

Exercici 10

Escriu en codi GLSL com calcular la posició de la càmera en *object space*. Es valorarà la eficiència.

vec3 obs = modelViewMatrixInverse[3] .xyz;

Més eficient que (modelViewMatrixInverse*vec4(0,0,0,1)).xyz;

Exercici 11

Indica clarament una tècnica utilitzada per calcular coordenades de textura per la qual **no** es pot assumir que les coordenades de textura (s,t) s'interpolen linealment en espai objecte. Explica la raó d'aquesta no linearitat.

Projective texture mapping. El motiu és que les coordenades de textura de cada punt es veuen afectades per la deformació de perspectiva associada al "projector".

Exercici 12

Indica en quin interval (el més ajustat possible) poden estar les coordenades Z dels punts interiors a la piràmide de visió d'una càmera perspectiva, segons l'espai considerat. Pots fer servir ±∞ si s'escau.

- Object space: (-∞,+∞)
  Eye space: (-∞,0)
  NDC: [-1,1]
  Window space:
  [0,1]

Exercici 13

En molts shaders sovint ens trobem amb la necessitat de transformar valors linealment d'un cert interval inicial a un cert interval de sortida. Per exemple, ens pot interessar transformar valors de l'interval [-1,1] a valors en l'interval [0,1].

Escriu una funció map() en GLSL que, donat un valor x dins l'interval [imin,imax], calculi el resultat interpolant linealment dins l'interval [omin, omax], de forma que al valor imin li correspongui el valor omin, i al valor imax li correspongui omax.

```
float map(float x, float imin, float imax, float omin, float omax)
{
flooat n = (xximin)/(i mmax-imin);// n dins[0,1]
ret   urn omax   + n*(omax-  omin);  // dins [omin, omaxx]




}
```

Exercici 14

Indica, per cadascun d'aquests modes de filtrat per MIGNIFICATION, quants texels es fan servir per avaluar cada crida a texture().

GL_NEAREST_MIPMAP_NEAREST 1

GL_LINEAR_MIPMAP_NEAREST 4

GL_NEAREST_MIPMAP_LINEAR

GL_LINEAR_MIPMAP_LINEAR 822

Exercici 15

Observa la següent figura. Indica, per cada tècnica, si l'ha pogut reproduir o no, **justificant cada resposta**.

(a) Bump mapping No, s'observen oclusions

(b) Parallax mapping: No, s'observen oclusions

(c) Relief mapping: Si, es plausible

(d) Displacement mapping: SI, es plausible

Exercici 16

Per una determinada primitiva, sabem les coordenades de textura que tindran alguns fragments:

Coordenades (x,y)	Coordenades (s,t)
100.5, 100.5	0.3, 0.1
101.5, 100.5	0.5, 0.8

Indica quin serà el resultat d'avaluar

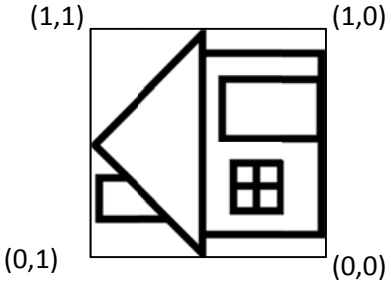
```
dFdx(texCoord)
```

per aquests fragments.

: (0.5, 0.8) – (0.2, 0.7) (→ (0.3, 0.1

Exercici 17

Amb la textura de l'esquerra hem texturat el quad de la dreta. Indica, al mateix quad, les coordenades de textura (s,t) de cada vèrtex.



Quin valors tindran dFdy(tex CCoord.s) i x(texCoord.t) als fragments del quad anterior?

dFdy((texCoord.s) = dFdx(tex dFdd

Exercc ici 18 CCoord.t) = 0.

Escriviu codi en llenguatge GLSL 3.30 core per passar un punt P de *clip space* a *model space*

```
vec4 P;
...
    P = modeelViewProjectionMatrixInverse * P;
```