Examen Parcial de Gràfics Curs 2011-12 Q2

Nom i Cognoms:

Exercici 1 *(1 punt)*

Considerant les següents definicions,

```
vec4 P;     // coordenades homogènies d'un punt
vec3 N;     // components d'un vector normal
vec3 L;     // components d'un vector cap a la font de llum
```

Escriviu el codi en llenguatge GLSL per fer les següents conversions:

(a) Passar P de *model space* a un vec3 en *normalized device space*

```
P = gl_ModelViewProjectionMatrix * P;
vec3 R = P.xyz / P.w;
```

(b) Passar P de *eye space* a *model space*

```
vec4 R = gl_ModelViewMatrixInverse * P;
```

(c) Passar N de *eye space* a *model space*

```
vec3 R = gl_NormalMatrixInverse * N;   // però donat que NormalMatrixInverse no existeix...
vec3 R = (vec4(N, 0.0) * gl_ModelViewMatrix).xyz; // ja que ((M^-1)^T)^-1*N  =  M^T*N = N*M
vec3 R = (gl_ModelViewMatrixTranspose * vec4(N, 0.0) ).xyz;
```

Exercici 2 *(1 punt)*

Escriu el codi GLSL per calcular (en un vertex shader), un vec3 amb la posició de l'observador en *object space,* sense fer servir cap uniform ni cap atribut.

```
vec3 obs = (gl_ ModelViewMatrixInverse * vec4(0.0, 0.0, 0.0, 1.0)).xyz;
```

o directament

```
vec3 obs = gl_ ModelViewMatrixInverse[3].xyz;  // darrera columna
```

Exercici 3 *(0.5 punt)*

(a) Escriu el nom d'un algorisme d'il·luminació global dels estudiats a classe capaç de simular interreflexions difoses entre els objectes de l'escena.

Radiositat

(b) Escriu el nom d'un algorisme d'il·luminació global dels estudiats a classe capaç de simular interreflexions especulars entre els objectes de l'escena.

Raytracing

Exercici 4 *(1 punt)*

Com has de declarar els següents objectes en GLSL (incloent-hi els qualificadors const, attribute, uniform, varying... i el tipus)?

(a) Normal que es passa del vertex shader al fragment shader

varying vec3 normal;

(b) Velocitat amb la que volem que girin automàticament tots els objectes de l'escena

uniform float speed;

(c) Vector tangent d'un vèrtex que rebem de l'aplicació

attribute vec3 tangent;

(d) Textura 2D

uniform sampler2D sampler;

Exercici 5 *(1 punt)*

Indica clarament on té més sentit realitzar les següents tasques, al vertex shader (VS) o al fragment shader (FS):

(a) Transformar el vèrtex de model space a clip space

VS

(b) Calcular automàticament les coordenades de textura (s,t) segons dos plans S i T.

VS

(c) Multiplicar les coordenades de textura per la matriu GL_TEXTURE

VS

(d) Implementar bump mapping

FS

Exercici 6 *(1 punt)*

La crida gluPerspective(fovy, aspect, n, f) multiplica la matriu actual per la següent matriu:

$$P = \begin{bmatrix} \frac{\cot\frac{fovy}{2}}{aspect} & 0 & 0 & 0 \\ 0 & \cot\frac{fovy}{2} & 0 & 0 \\ 0 & 0 & \frac{n+f}{n-f} & \frac{2*n*f}{n-f} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

Suposant que la matriu ModelView és la identitat, i que V=(x,y,z,1) és un punt en model space:

(a) Indica quina serà la coordenada Z en clip space del punt V.

z(n+f)/(n-f) + 2nf/(n-f)

(b) Indica quin serà el valor de la Z en normalized device space?

(z(n+f)/(n-f) + 2nf/(n-f)) / -z

(c) La coordenada Z en normalized device space depèn **linealment** de les coordenades x, y, z en eye space? Òbviament no és una dependència lineal (sinó tindria la forma ax+by+cz; però la z apareix **dividint**)

Exercici 7 *(1 punt)*

Una forma rudimentària de simular boira d'un cert color és interpolant linealment el color del fragment amb el color de la boira, segons la coordenada z del fragment. Escriu l'expressió per calcular gl_FragColor utilitzant aquesta tècnica:

```
uniform vec4 colorBoira;
gl_FragColor = mix(gl_Color, colorBoira, gl_FragCoord.z);
```

Exercici 8 *(2 punts)*

La següent figura mostra la textura d'un tauler d'escacs (a l'esquerra) i una finestra OpenGL amb un polígon texturat amb aquesta textura (a la dreta):

El vèrtex inferior esquerra del polígon té coordenades de textura (0,0), i el vèrtex superior dret té coordenades de textura (1,1).

Si (s,t) denoten les coordenades de textura a cada fragment, i (x,y) són coordenades (en window space) dels fragments, responeu aquestes qüestions, justificant la resposta:

(a) Indica, a la imatge de la dreta, en quina regió es maximitza el valor de $\frac{\partial s}{\partial x}$

El valor de $\frac{\partial s}{\partial x}$ es maximitza a l'aresta superior del polígon (la longitud de les cel·les del tauler a l'aresta superior és aproximadament la meitat de la longitud de les cel·les del tauler a l'aresta inferior, el que indica que $\frac{\partial s}{\partial x}$ a l'aresta superior és aproximadament el doble que $\frac{\partial s}{\partial x}$ a l'aresta inferior.

(b) Indica, a la imatge de la dreta, en quina regió es maximitza el valor de $\frac{\partial t}{\partial y}$

El valor de $\frac{\partial t}{\partial y}$ es maximitza també a l'aresta superior del polígon (compareu l'alçada de les cel·les del tauler amunt i avall).

(c) Què podeu dir del valor de $\frac{\partial t}{\partial x}$? És zero a tots els fragments: tots els fragments d'una mateixa línea tenen la mateixa coordenada de textura t. Això fa que les diferents línies de cel·les es vegin horitzontals a la finestra OpenGL.

Exercici 9 *(1 punt)*

Indica, per cadascun d'aquests mètodes de filtrat, quants texels cal consultar per calcular el color de la mostra.

(a) GL_NEAREST 1
(b) GL_LINEAR 4
(c) GL_NEAREST_MIPMAP_NEAREST 1
(d) GL_LINEAR_MIPMAP_NEAREST 4
(e) GL_NEAREST_MIPMAP_LINEAR 2
(f) GL_LINEAR_MIPMAP_LINEAR 8

Exercici 10 *(0.5 punt)*

Indica quines d'aquestes prestacions són suportades per la tècnica de bump mapping estudiada a classe.

(a) Variacions d'il·luminació en superfícies rugoses SI

(b) Self-occlusion NO

(c) View parallax NO

(d) Detalls (rugositat) a la silueta dels objectes NO