Examen Parcial de Gràfics Curs 2012-13 Q1

Nom i Cognoms:

Exercici 1 *(2 punts)*

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

(a) Passar el punt Pclip de *clip space* a *object space*

```
vec4 Pclip, Pobj;
...
Pobj = gl_ModelViewProjectionMatrixInverse * Pclip;
```

(b) Passar el punt Pwin de *window space* a *normalized device space*; només ens interessen les coordenades x,y.

```
float width, height;  // mida del viewport; podeu assumir que el viewport comença al (0,0)
vec4 Pwin;
...
vec2 Pndc =  2.0*vec2(Pwin.x/width, Pwin.y/height) – vec2(1.0);
```

(c) Passar una normal N de object space a eye space

```
vec3 N;
...
N = gl_NormalMatrix * N;
```

(d) Passar un punt P de *eye space* a *world space*.

```
uniform mat4 modelingTransform; // matriu de modelat de l'objecte
vec4 P;
...
P = modelingTransform * gl_ModelViewMatrixInverse * P;
```

Exercici 2 *(0.5 punt)*

Escriu el codi GLSL per calcular (en un vertex shader), un vec3 amb la posició de l'observador en *clip space,* sense fer servir cap variable definida per l'usuari.

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

Exercici 3 *(1 punt)*

Completa el codi GLSL de sota per obtenir la coordenada z d'un fragment en *eye space*.

Pista 1: podeu suposar que la matriu gl_ProjectionMatrix conté la matriu que genera la crida gluPerspective(fovy, aspect, n, f), que és:

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

Pista 2: gl_FragCoord.w conté el valor $1/w_{clip}$, és a dir, la inversa de la coordenada homogènia en clip space.

```
vec4 gl_FragCoord;
float zeye = -1.0 / gl_FragCoord.w;
```

Observeu que:

gl_FragCoord.w = $1/w_{clip}$

w_{clip} =(gl_ProjectionMatrix * P).w = –P.z (P en eye space)

Per tant, gl_FragCoord.w = -1/P.z.

Exercici 4 *(0.5 punt)*

Volem visualitzar amb un cert nivell de realisme el model 3D d'un aula com la del A4202, un dia ennuvolat, amb els llums tancats. Quina mena de model d'il·luminació necessitem i per què?

Necessitem un model d'il·luminació global, perquè tota la llum que rep l'escena és llum indirecta.

Exercici 5 *(1 punt)*

Completa aquests shaders per obtenir de forma alternativa la coordenada z d'un fragment en eye space.

Vertex shader

```
varying float z;
void main() {
    z = (gl_ModelViewMatrix * gl_Vertex).z;
```

Fragment shader

```
varying float z; // z en eye space
```

Exercici 6 *(1 punt)*

Indica, per les següents accions, si són factibles (SI) o no (NO) dins un fragment shader:

(a) Modificar les coordenades x,y del fragment

NO

(b) Descartar el fragment

SI

(c) Consultar el valor del depth buffer al píxel corresponent al fragment.

NO

(d) Implementar bump mapping

SI

Exercici 7 *(1 punt)*

Sovint és útil retallar la geometria de la nostra escena amb plans de retallat addicionals als sis plans que delimiten la piràmide de visió. Completeu el següent fragment shader per tal de simular l'efecte d'un pla de retallat addicional que elimini tot el que queda per sota del pla y=0.0 (en eye space).

```
varying vec4 pos; // posició en object space

void main() {
        if  ( (gl_ModelViewMatrix * pos).y < 0.0)
                discard;
        gl_FragColor = gl_Color;
}
```

Exercici 8 *(1 punt)*

Sabem que, en una aplicació concreta de museus virtuals, la distància de la càmera als objectes texturats és tal que la pre-imatge d'un fragment té una mida sempre inferior a un texel.

(a) Quina mena de filtrat (magnification o minification) necessitem?
Magnification

(b) Recomanaries fer servir mipmapping? NO, no té sentit.

Exercici 9 *(1 punt)*

Suposeu que P(u,v) és una superfície parametritzada. Indica quina interpretació geomètrica tenen els següents vectors:

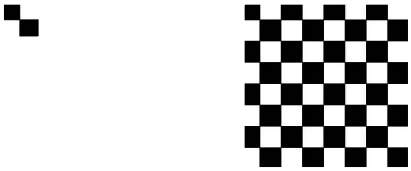
(a) ∂P/∂u Vector tangent a la superfície en direcció paramètrica u

(b) ∂P/∂v Vector tangent a la superfície en direcció paramètrica v

(c) ∂P/∂u x ∂P/∂v Vector normal a la superfície

Exercici 10 *(1 punt)*

Disposem de la textura de 2x2 texels (mostrada a l'esquerra de la figura), amb la que volem texturar un polígon per tal que sembli un tauler d'escacs, com mostra la dreta de la figura:



(a) Indica, al mateix polígon, les coordenades de textura (s,t) que hem d'utilitzar per cada vèrtex.
En ordre antihorari, començant pel vèrtex de la cantonada inferior esquerra:
(0,0), (4,0), (4,4), (0,4).

(b) Quin mode de wrapping (GL_CLAMP, GL_CLAMP_TO_EDGE, GL_REPEAT) hem de fer servir?
GL_REPEAT

(c) Quin magnification filter recomanes en aquest cas? GL_NEAREST