Examen Final de Gràfics Curs 2014-15 Q2

Nom i Cognoms:

Tots els exercicis tenen el mateix pes.

Exercici 1

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. Suposa que només hi ha un VS i un FS (no hi ha GS).

- Escritura de gl_Position	Escritura de gl_Position
- Rasterització	Retallat de primitives (clipping)
- Retallat de primitives (clipping)	Rasterització
- Stencil test	Stencil test

Exercici 2

Indica, per cada etapa de la llista de sota, si és ANTERIOR o POSTERIOR a la etapa de rasterització:

(a) Primitive assembly ANTERIOR

(b) Processament del fragment POSTERIOR

(c) Ús de les funcions dFdX, dFdY POSTERIOR

(d) Pas de les coordenades a clip space ANTERIOR

Exercici 3

Indica al menys una tasca típica al pipeline de visualització programable, que es pugui fer tant abans com després de la rasterització.

Càlculs d'il·luminació; generació de coordenades de textura (sphere mapping...), ...

Exercici 4

Hem produït un fragment amb color RGBA = (1.0, 0.5, 0.0, 0.8) corresponent al pixel (i,j). El color RGBA del pixel (i,j) al buffer de color és (1.0, 0.5, 1.0, 1.0). Si tenim activat alpha blending amb la crida

glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA)

quin serà el color RGBA resultant al buffer de color (assumint que passa tots els tests)?

0.8*1.0 + 0.2*1.0 = 1.0

0.8*0.5 + 0.2*0.5 = 0.5

0.8*0.0 + 0.2*1.0 = 0.2

0.8*0.8 + 0.2*1.0 = 0.84 → (1.0, 0.5, 0.2, 0.84)

Exercici 5

Indica quina és la matriu que aconsegueix la conversió demanada (assumint que no hi ha transformació de modelat), usant aquestes abreviatures:

MV = gl_ModelViewMatrix MVP = gl_ModelViewProjectionMatrix
MV^{-1} = gl_ModelViewMatrixInverse MVP^{-1} = gl_ModelViewProjectionMatrixInverse
NM = gl_NormalMatrix I = Identitat

a) Pas d'un vèrtex de eye space a world space MV^{-1}

b) Pas d'un vèrtex de clip space a object space MVP^{-1}

c) Pas de la normal de object space a eye space NM

d) Pas d'un vèrtex de object space a world space I

Exercici 6

En una hipotètica versió de GLSL, no està definit el uniform **gl_ModelViewProjectionMatrixInverse** (però sí la resta de matrius de GLSL). Escriu, en codi GLSL vàlid, com faries la conversió de **vec4 P** de clip space a object space (assumint que no hi ha transformació de modelat).

gl_ModelViewProjectionMatrixInverse * P;

->

gl_ModelViewMatrixInverse * gl_ProjectionMatrixInverse * P;

Exercici 7

Completa el següent FS per tal que calculi correctament el vector L que intervé a la contribució

difosa: vec4 lightSource(vec3 N, vec3 P, gl_LightSourceParameters light) {
// N és la normal en eye
space // P és el punt en eye
space

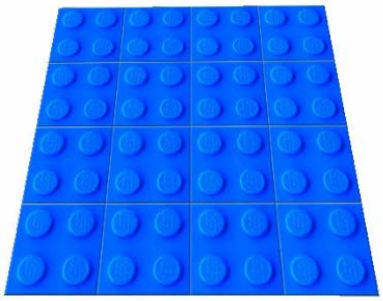
vec3 L = normalize(light.position.xyz - P);

float diff = max(0.0, dot(N,L));
return gl_FrontMaterial.diffuse * light.diffuse *
diff;

Exercici 8

Amb la textura de l'esquerra, volem texturar l'objecte Plane com a la

dreta.



Completa la línia que necessitem al VS:

gl_TexCoord[0].st = 4.0 * gl_MultiTexCoord0.st;

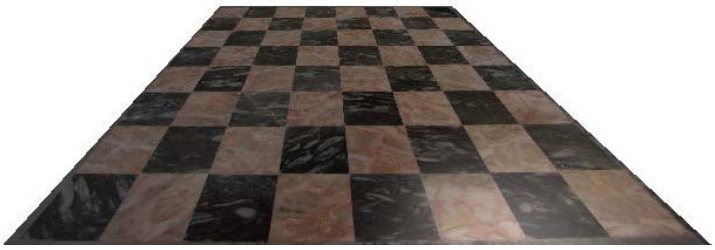
Exercici 9

Quants nivells de detall (nivells de LOD) formen la piràmide mipmapping completa d'una textura de 256x256 texels (tenint en compte el LOD 0)?

1+log2(256) = 9 (256x256, 128x128, 64x64, 32x32, 16x16, 8x8, 4x4, 2x2, 1x1)

Exercici 10

La següent figura mostra la textura d'un tauler d'escacs (a l'esquerra) i el Plane texturat (a la dreta):



El vèrtex inferior esquerra del polígon té coordenades de textura (0,0), i el vèrtex superior dret (1,1). Indica, a la imatge de la dreta, en quina regió es maximitza el valor de $\frac{\partial \underline{s}}{\partial x}$

Es maximitza a l'aresta superior del polígon.

Exercici 11

Tenim un model amb **coordenades de textura 2D que cobreixen l'interval [-1,1].** Ens demanen un VS que mostri el model projectat sobre aquest espai de textura (com a l'exercici UVunfold), de forma que ocupi tot el viewport, amb independència de la càmera. Completa aquesta línia del VS demanat:

```
gl_Position = vec4( gl_TexCoord[0].s, gl_TexCoord[0].t, 0.0, 1.0);
```

Exercici 12

Imagina un FS al qual li arriben les coordenades del punt (**vec4 P**) en un cert espai de coordenades.

Indica quin test hauries de fer sobre **P** per tal **d'eliminar (discard) els fragments que cauen a la meitat dreta** de la finestra OpenGL, amb mida 800x600:

a) Si P està en clip space

```
if ( P.x/P.w > 0 ) discard;
```

b) Si P està en NDC

```
if ( P.x > 0 ) discard;
```

c) Si P està en window space

```
if ( P.x > 400 ) discard;
```

Exercici 13

Escriu quina matriu 4x4 necessitem per simular la reflexió de l'escena respecte un mirall situat al pla

Y=0.1

```
0  0  0
0  -1 0  0
0  0  1  0
0  0  0  1
```

Exercici 14

Tenim un cub format per 6 cares i 8 vèrtexs.

a) Si volem que cada corner (vèrtex associat a una única cara) tingui la normal de la cara a la que pertany, quants vèrtexs necessitem en el VBO del cub?

6 cares * 4 vèrtexs per cada cara = **24 vèrtexs**

b) Si volem que cada vèrtex del cub tingui com a normal el promig de les normals de les cares que incideixen al vèrtex, quants vèrtexs necessitem (com a mínim) en el VBO del cub?

8 vèrtexs (mínim)

Exercici 15

Tenim un normal map on les components RGB de cada texel codifiquen les components (nx',ny',nz') en espai tangent d'un vector unitari en la direcció de la normal pertorbada. Donats els vectors T,B,N (tangent, bitangent i normal) d'una base ortonormal, en eye space, corresponents a un fragment, indica com calcularies la normal pertorbada N' en eye space.

Simplement hem de passar la normal (nx',ny',nz') de tangent space a eye space: [T B N] * (nx',ny',nz') on [T B N] és una matriu 3x3 que té per columnes els tres vectors T, B, N.

Exercici 16

Quins són els sistemes de coordenades origen i destí de la transformació representada per la gl_ModelViewMatrix en GLSL?

Object Space --> Eye Space

Exercici 17

Aquesta és la declaració de la funció OpenGL glDepthMask():

```
void glDepthMask( GLboolean foo)
```

Quin efecte té aquesta funció?

Activar (true) / desactivar (false) l'escriptura en el depth buffer.

Exercici 18

De quin tipus GLSL (float, vec2, vec3...) és el resultat d'aquestes expressions?

a) dot(vec3(1,0,0), vec3(0,1,0))

float

b) cross (vec3(1,0,0,), vec3(0, 1, 0))

vec3

Exercici 19

Indica quantes vegades cal pintar l'escena en les següents tècniques:

a) Shadow mapping, suposant que la llum és dinàmica 2 cops

b) Ombres per projecció, versió sense stencil 2 cops

c) Reflexió amb objectes virtuals, amb stencil (ignoreu els passos en que només es dibuixa el mirall): 2 cops

d) Projective Texture Mapping, amb una textura fixa 1 cop

Exercici 20

Tenim una aplicació amb una escena opaca, sense miralls, amb una única font de llum puntual situada al punt (0,0,0) en coordenades de l'observador. Quina tècnica creus apropiada per reproduir les ombres en aquest cas?

Cap; en aquest cas, la càmera coincideix amb la llum, i per tant les ombres no seran visibles.

CT, MR, ultrasons...