

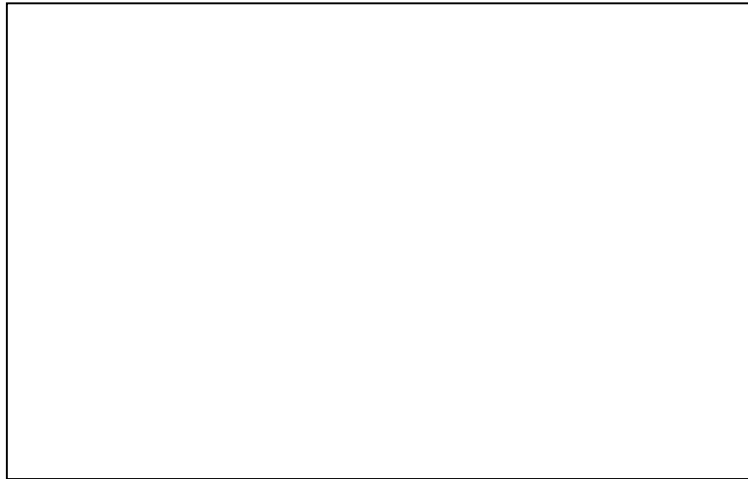
Nom i Cognoms:

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).

- Divisió de perspectiva
- Escriptura de `gl_Position`
- `glDrawElements`
- Rasterització



Exercici 2

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

- (a) Geometry Shader
- (b) Fragment Shader
- (c) Stencil Test
- (d) Alpha blending

Exercici 3

Quants nivells de detall (nivells de LOD) formen la piràmide mipmapping completa d'una textura de 512x512 texels (tenint en compte el LOD 0)?

Exercici 4

Tenim un octaedre representat amb una malla triangular formada per 6 vèrtexs i 8 triangles. Volem construir un VBO per representar aquest octaedre, de forma que el VS rebi com a atributs les coordenades (x,y,z) del vèrtex i les components del vector normal (nx,ny,nz), **sense cap suavitzat d'aresta** (volem que l'octaedre aparegui il·luminat correctament). Cada coordenada/component estarà representada per un float (4 bytes).

(a) Quants vèrtexs necessitem representar al VBO?

(b) Quants índexs (*elements*) es necessiten a l'array d'índexs?

(c) Quina mida (en bytes) tindrà l'array de coordenades de vèrtexs?

Exercicis 5 i 6

Indica quina és la matriu (o **producte de matrius**) que aconseguirà la conversió demanada, usant la notació següent (vigileu amb l'ordre en que multipliqueu les matrius):

M = modelMatrix	M^{-1} = modelMatrixInverse
V = viewingMatrix	V^{-1} = viewingMatrixInverse
P = projectionMatrix	P^{-1} = projectionMatrixInverse
N = normalMatrix	I = Identitat

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

b) Pas d'un vèrtex de object space a eye space

c) Pas d'un vèrtex de eye space a clip space

d) Pas d'un vèrtex de eye space a world space

A) Pas d'un vèrtex de clip space a object space

B) Pas d'un vèrtex de world space a clip space

C) Pas d'un vèrtex de object space a model space

D) Pas de la normal de object space a eye space

Exercici 7

Tenim activat alpha blending amb la crida

```
glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA)
```

Hem produït un fragment amb color RGBA = (1.0, 0.5, 0.0, **a**) corresponent al pixel (i,j). El color RGBA del pixel (i,j) al buffer de color és (1.0, 1.0, 0.0, 0.0). Quin valor ha de tenir **a** si volem que les components RGB del resultat siguin (1.0, 0.8, 0.0)?

Exercici 8

Indica quina és la diferència més important entre els models d'il·luminació local i els models d'il·luminació global.

Exercici 9

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

```
uniform vec4 matDiffuse;
uniform vec4 lightDiffuse, lightPosition;

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

    vec3 L =

    float diff = max( 0.0, dot( N,L ) );
    return matDiffuse * lightDiffuse * diff;
}
```

Exercici 10

Indica, en la notació estudiada a classe, $L(D|S)*E$, quins light paths són suportats per ray-tracing clàssic.

Exercici 11

Volem generar amb RayTracing una imatge 1024x1024 d'una escena interior tancada. Tots els objectes són opacs i perfectament difosos, i hi ha quatre fonts de llum.

a) Quants rajos primaris caldrà traçar?

b) Quants shadow rays caldrà traçar, en total?

c) Quants rajos reflectits caldrà traçar, en total?

d) Quants rajos transmesos caldrà traçar, en total?

Exercici 12

Relaciona cada concepte de l'esquerra amb un concepte de la dreta:

- | | |
|----------------|---|
| 1. Flux | A. ϕ per unitat d'angle sòlid |
| 2. Intensitat | B. Mesurat en lúmens |
| 3. Irradiància | C. Energia (per unitat de temps) que travessa un punt en una determinada direcció |
| 4. Radiància | D. Mesurat en lux |

1 →

2 →

3 →

4 →

Exercici 13

Completa l'equació general del rendering:

$$L_o(x, \omega_o) = L_e(x, \omega_o) + \int f(x, \omega_i, \omega_o) L_i(x, \omega_i) \boxed{} d\omega_i$$

Exercici 14

Quin és el nom de l'equació que ens permet saber la quantitat de llum reflectida i la quantitat de llum transmesa, en funció de diferents paràmetres (angle incident, etc)?

Exercici 15

Volem dibuixar una escena amb alguns objectes opacs i altres translúcids, amb alpha blending. Donat que no volem ordenar els objectes, fem servir aquest codi:

```
glEnable(GL_BLEND);
glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);

if (depthTestOpaque) glEnable(GL_DEPTH_TEST);
else glDisable(GL_DEPTH_TEST);
glDepthMask(maskOpaque);
drawObjects(opaqueObjects);

if (depthTestTranslucent) glEnable(GL_DEPTH_TEST);
else glDisable(GL_DEPTH_TEST);
glDepthMask(maskTranslucent);
drawObjects(translucentObjects);
```

Indica els valors més adients per les variables booleanes:

- (a) depthTestOpaque
- (b) depthTestTranslucent
- (c) maskOpaque
- (d) maskTranslucent

Exercici 16

Tenim una escena amb diferents objectes, cadascú amb la seva transformació de modelat. Volem simular les reflexions en un mirall pla fent servir la tècnica de reflexió amb objectes virtuals. Hem calculat la matriu R de reflexió respecte el pla del mirall, usant els coeficients (a,b,c,d) del pla en world space. Fent servir la notació

M = modelMatrix (original) M^{-1} = modelMatrixInverse (original)
 V = viewingMatrix V^{-1} = viewingMatrixInverse

indica quin producte de matrius és necessari **en el pas en que es dibuixen els objectes en posició virtual**:

- (a) Passar un vèrtex de object space a eye space
- (b) Passar un vèrtex de object space a world space

Exercici 17

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:

```
layout (location = 3) in vec2 texCoord;
```

```
gl_Position = vec4( , , 0.0, );
```

Exercici 18

Tenim aquest fragment de FS:

```
void main() {  
    vec3 I = normalize(Pos);  
    vec3 R = reflect(I, N);  
    fragColor = sampleTexture(R);  
}
```

(a) Quina tècnica està implementant?

(b) En quin espai de coordenades està treballant?

Exercici 19 i 20

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

a) Shadow mapping, suposant que la llum és dinàmica

b) Ombres per projecció, versió sense stencil

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

d) Environment Mapping, amb una textura fixa

Preguntes per a l'avaluació de les competències transversals

Pregunta 1

Explica què és i per què es fa servir la funció de transferència (transfer function) en aplicacions mèdiques

Pregunta 2

Explica per què serveix la tècnica de *Image Registration*, dins les aplicacions dels gràfics en medicina.