

48130.pdf



fibsbook



Gráficos



3º Grado en Ingeniería Informática



Facultad de Informática de Barcelona (FIB)
Universidad Politécnica de Catalunya

EAE Business School
Barcelona

MÁSTER EN PROJECT MANAGEMENT

Convocatoria Abril 2023

eaebarcelona.com

Work
to change
your life

Elige tu propio camino
y empieza a cambiar lo
que tú quieras cambiar.

We make
it happen



CONOCE EL MÁSTER

Encuentra **el trabajo** de tus sueños

Participa en retos y competiciones de programación



Escanéame y
obtén más info!!

Examen Final de Gràfics

Curs 2018-19 Q1

Nom:

Exercici 1

Copia a la dreta aquestes quatre tasques del pipeline gràfic, però ordenades d'acord amb l'ordre d'execució. Pots assumir VS+FS.

- Perspective Division
- glDrawElements
- Rasterization
- Stencil test

Exercici 2

Copia a la dreta aquestes quatre tasques del pipeline gràfic, però ordenades d'acord amb l'ordre d'execució. Pots assumir VS+FS.

- Alpha blending
- glBufferData
- Rasterization
- Write to gl_Position

Exercici 3

Escriu una condició necessària (per a la llum) per a que les ombres d'una escena tinguin una zona de penombra.



Exercici 4

Escriu, per cada tasca, si bé ABANS o DESPRÉS de la rasterització:

- (a) Pas a NDC
- (b) Geometry Shader
- (c) discard
- (d) Depth Test

Exercici 5

Per a què serveix la funció **glPolygonOffset**? Indica clarament què dades del fragment modifica.

Exercici 6

Completa aquest codi que correspon al primer pas de l'algorisme d'ombres per projecció:

```
// Draw receiver onto stencil buffer
glEnable(GL_STENCIL_TEST);
glStencilFunc(          , 1, 1);
glStencilOp(GL_KEEP, GL_KEEP,          );
draw(receiver);
```

Exercici 7

Escriu, usant la notació $L(D|S)*E$, els light paths que suporten aquestes tècniques:

- (a) Two-pass raytracing
- (b) Classic Raytracing

Exercici 8

Quin concepte de radiometria/fotometria és el més adient per mesura la quantitat d'energia per unitat de temps que arriba a una superfície, per unitat d'àrea (unitats W/m^2)?

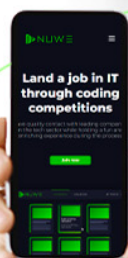
Encuentra **el trabajo** de tus sueños

Participa en retos y competiciones de programación

Ten contacto de calidad con empresas líderes en el sector tecnológico mientras vives una experiencia divertida y enriquecedora durante el proceso.

Únete ahora

Escanéame y
obtén más info!!



Exercici 9

Aquest VS calcula coordenades de textura projectives per a un FS que implementa shadow mapping:

```
uniform mat4 lightMatrix;  
out vec4 textureCoords;  
...  
void main() {  
    ...  
    textureCoords = lightMatrix*vec4(vertex,1);  
    gl_Position = modelViewProjectionMatrix *vec4(vertex,1);  
}
```

Usant aquesta notació:

S(sx,sy,sz) -> Scale matrix	T(tx,ty,tz) -> Translate matrix
M -> model matrix (of the object)	V -> view matrix (of the light camera)
P -> projection matrix (of the light camera)	

Escriu (com a producte de matrius) com l'aplicació ha de calcular la matriu pel uniform **lightMatrix**.

Exercici 10

Escriu la matriu o producte de matrius per les conversions següents, usant la notació:

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

- a) Convertir un vèrtex de world space a clip space
- b) Convertir un vèrtex de eye space a clip space
- c) Convertir un vèrtex de eye space a world space
- d) Convertir una normal de object space a eye space

Encuentra el trabajo de tus sueños

Participa en retos y competiciones de programación



Escanéame y obtén más info!!

Exercici 11

A l'equació general del rendering:

$$L_o(\mathbf{x}, \omega_o, \lambda, t) = L_e(\mathbf{x}, \omega_o, \lambda, t) + \int_{\Omega} f_r(\mathbf{x}, \omega_i, \omega_o, \lambda, t) L_i(\mathbf{x}, \omega_i, \lambda, t) (\omega_i \cdot \mathbf{n}) d\omega_i$$

(a) Què és Ω ?

(b) Quin vector juga el paper del light vector L que s'utilitza al model d'il·luminació de Lambert?

Exercici 12

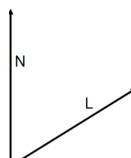
Completa aquest FS per calcular Phong lighting:

```
uniform vec4 matAmbient, matDiffuse, matSpecular;  
uniform vec4 lightAmbient, lightDiffuse, lightSpecular, lightPosition;  
uniform float matShininess;
```

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

Exercici 13

Dibuixa el vector calculat per l'expressió $2(\mathbf{N} \cdot \mathbf{L})\mathbf{N} - \mathbf{L}$



Exercici 14

Dóna un exemple de tècnica que requereixi dibuixar les primitives de l'escena ordenades back to front respecte la càmera.

Exercici 15

En quin sistema de coordenades han d'estar aquestes variables per tal que el VS sigui correcte?

(a) `vec3 L = normalize(lightPosition.xyz - P); // L is the light vector`

P ha d'estar en espai _____

(b) `gl_Position = projectionMatrix * P;`

P ha d'estar en espai _____

Exercici 16

Aquests fragments de codi tenen un o més errors. Re-escriu-los de forma correcta:

(a) `vec4 P = modelMatrix * viewMatrix * projectionMatrix * vec4(vertex, 1.0);`

(b) `vec3 N = modelViewMatrix * vec4(normal,0);`