

FinalG2019-2020Q1Soluciones.pdf



Llull179



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

1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

BBVA está adherido al Fondo de Garantía de Depósitos de Entidades de Crédito de España. La cantidad máxima garantizada es de 100.000 euros por la totalidad de los depósitos constituidos en BBVA por persona.

1
Abre tu Cuenta Online sin comisiones ni condiciones

2
Haz una compra igual o superior a 15€ con tu nueva tarjeta

3
BBVA te devuelve un máximo de 15€

¿Cómo?

Cuéntame más



Examen Final de Gràfics

Curs 2019-20 Q1

Nom:

Exercici 1

Copia a la dreta aquestes quatre les tasques del *pipeline* gràfic, però ordenades d'acord amb l'ordre d'execució.

- Alpha Blending
- Fragment shader
- Geometry shader
- Rasterització

Exercici 2

Copia a la dreta aquestes quatre les tasques del *pipeline* gràfic, però ordenades d'acord amb l'ordre d'execució.

- Depth test
- glDrawElements
- Stencil Test
- Vertex Shader

Exercici 3

Escriu quin és l'espai de coordenades inicial i final de la multiplicació de la **modelViewProjectionMatrixInverse** per un vèrtex.

Inicial:

Final:

Exercici 4

Escriu, per cada tasca, en quin o quins shaders (VS, GS, FS) és possible:

- (a) discard
- (b) EndPrimitive()
- (c) Escriure gl_Position
- (d) dFdx, dFxy

Exercici 5

Què coordenada del fragment modifica la funció **glPolygonOffset**?

En quin espai la modifica?

Exercici 6

Indica, amb la notació vista a classe, el *light path* que explica el color dominant del píxel indicat a la imatge.



Exercici 7

Indica, per cada punt, si pot ser dins (DINS) o segur que no (FORA) la piràmide de visió d'una càmera perspectiva. Escriu una breu explicació.

- (a) (0.2, -1.5, 1.8, 2) en clip space
- (b) (0, 0, 10, 1) en object space



WUOLAH + BBVA

1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

BBVA está adherido al Fondo de Garantía de Depósitos de Entidades de Crédito de España. La cantidad máxima garantizada es de 100.000 euros por la totalidad de los depósitos constituidos en BBVA por persona.

Te regalamos

15€

1
Abre tu Cuenta Online sin comisiones ni condiciones

2
Haz una compra igual o superior a 15€ con tu nueva tarjeta

3
BBVA te devuelve un máximo de 15€

¿Cómo? →

Cuéntame más



WUOLAH
+ BBVA

Exercici 8

Re-escriu el codi GLSL subratllat amb una versió equivalent però més compacte:

```
float t;  
vec3 color1, color2;  
vec3 color = t*color1 + (1-t)*color2;
```

Exercici 9

Re-escriu aquest codi GLSL amb una versió equivalent més compacte:

```
vec3 obs = (modelViewMatrixInverse * vec4(0,0,0,1)).xyz;
```

Exercici 10

Escriu un exemple d'algorisme que suporti els *light paths* que s'indiquen:

(a) LS*DS*E (i LS*E)

(b) L(D|S)*E

Exercici 11



Volem aplicar la textura a un quad que té coordenades de textura inicials en [0,1].



Completa el FS per aconseguir el resultat que es mostra:

```
frontColor = texture(colorMap, _____ * vtexCoord);
```

1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

BBVA está adherido al Fondo de Garantía de Depósitos de Entidades de Crédito de España. La cantidad máxima garantizada es de 100.000 euros por la totalidad de los depósitos constituidos en BBVA por persona.

Exercici 12

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

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

Usant aquesta notació:

S(sx,sy,sz) -> Scale matrix
M -> model matrix (of the object)
P -> projection matrix (of the light camera)

T(tx,ty,tz) -> Translate matrix
V -> view matrix (of the light camera)

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

1
Abre tu Cuenta Online sin comisiones ni condiciones

2
Haz una compra igual o superior a 15€ con tu nueva tarjeta

3
BBVA te devuelve un máximo de 15€

Exercici 13

A l'equació general del rendering:

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

Què representa Ω ?

¿Cómo?

Exercici 14

Escriu la matriu o producte de matrius per convertir un vèrtex de *object space* a *eye space*, usant únicament les matrius que s'indiquen (no en falta cap per aquest exercici):

modelMatrix
projectionMatrix
modelViewProjectionMatrix

modelMatrixInverse
projectionMatrixInverse
modelViewProjectionMatrixInverse

Cuéntame más



Exercici 15

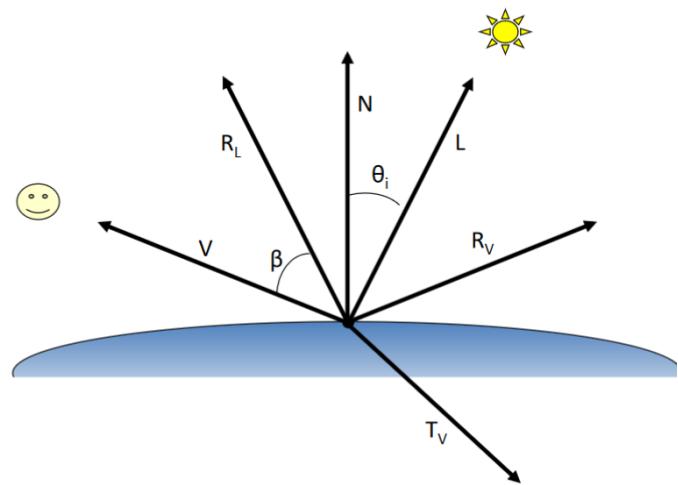
Indica quin tipus GLSL (float, vec2, vec3...) usaries per cada cas:

- (a) Segon paràmetre d'una crida a texture(colormap...)
- (b) Coordenades de textura que passa VS a FS en *shadow mapping*
- (c) Coordenades de textura que passa VS a FS en *projective texture mapping*
- (d) Vector per accedir a un *sphere map*

Exercici 16

Amb la notació de la figura, indica, en el cas de Ray-tracing

- (a) Quin vector té la direcció del *shadow ray*?
- (b) Quin vector és paral·lel al raig transmès?



Exercici 17

Continuant amb la figura anterior...

- (a) Quin vector té la direcció del raig primari?
- (b) Quins dos vectors determinen la contribució local de Phong?

Exercici 18

Quines són les unitats de la radiància (radiometria) en el Sistema Internacional?

Exercici 19

Completa, a sota, el codi que falta.

```
funció traçar_raig(raig, escena, μ)
    si profunditat_correcta() llavors
        info:=calcula_interseccio(raig, escena)
        si info.hi_ha_interseccio() llavors
            color:=calcular_ID(info,escena); // ID
            si es_reflector(info.obj) llavors
                raigR:=calcula_raig_reflectit(info, raig)
                color+= KR*traçar_raig(raigR, escena, μ) //IR
            fsi
            si es_transparent(info.obj) llavors
                [REDACTED]
            fsi
        sino color:=colorDeFons
        fsi
    sino color:=Color(0,0,0); // o colorDeFons
    fsi
    retorna color
ffunció
```

Exercici 20

Completa aquest fragment shader que implementa la tècnica de Shadow mapping:

```
uniform sampler2D shadowMap;
uniform vec3 lightPos;
in vec3 N,P;
in vec4 vtexCoord; // coordenades de textura en espai homogeni
out vec4 fragColor;

void main()
{
    [REDACTED]

    float NdotL = max(0.0, dot(N,L));
    vec4 color = vec4(NdotL);
    vec2 st = vtexCoord.st / vtexCoord.q;

    float storedDepth =
    [REDACTED]

    float trueDepth = vtexCoord.p / vtexCoord.q;
    if (trueDepth <= storedDepth) fragColor = color;
    else fragColor = vec4(0);
}
```

1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

BBVA está adherido al Fondo de Garantía de Depósitos de Entidades de Crédito de España. La cantidad máxima garantizada es de 100.000 euros por la totalidad de los depósitos constituidos en BBVA por persona.

2019-2020-Q1

1
Abre tu Cuenta Online sin comisiones ni condiciones

2
Haz una compra igual o superior a 15€ con tu nueva tarjeta

3
BBVA te devuelve un máximo de 15€

- ① Geometry shader → Rasterización → Fragment Shader → Alpha Blending
- ② ~~vertex shader~~ → glDrawElements → Vertex Shader → Depth Test → Stencil test
- ③ Inicial: Clip space Final: Object Space
- ④ a)iscard FS
b)EndPrimitive GS
- ⑤ Determina un offset por la proporcionalidad, es a dir, modifica ~~las coordenadas~~ las coordenadas \vec{z} en eye space
- ⑥ LSSE
- ⑦ a) $[0, -15, 18, 2]$ Dins ja que las coordenadas x, y, z están dentro del intervalo definido por S.C.C valor positivo e negativo de la coordenada homogénea $\Rightarrow -2 \leq [0, 15, 18] \leq 2$
b) $[0, 9, 10, 1]$ Dins ja que S.C.O
- ⑧ mix color = mix(color1, color2, t)
- ⑨ vec3 obs = cameraPosition
- ⑩ a) LS*DS*E ($: LS*E$) → Two Pass ray-tracing
b) L(DS)*E → Path Tracing
- ⑪ frontColor = texture(colorMap, vec2(~~0.5~~, 1, DS)) * vertexColor
- ⑫ P*V
- ⑬ Representa todos los direcciones de la esfera
- ⑭ modelViewProjection Matrix * projection Matrix Inverse
- ⑮ a) vec2
b) vec3
c) vec3
d) vec2
- ⑯ a) L, ja que es el que va del punto al ~~camara~~ para comprobar si cae a la sombra
b) V
- ⑰ a) V
b) El vector reflectante R_v i el vector de transmisión o refracción T_v
- ⑱ $\frac{W}{sr \cdot m^2}$



⑯ $r_{ray} T = \text{calcula_transmisi}\ddot{\text{o}} \text{ (info, ray)} \quad \cancel{\text{color}}$

color + = $k_T \cdot \text{trazarRay}(\text{ray}, \text{escena}, \mu)$

• $\text{vec3 } L = \text{lightPos.xyz} - P$

• ~~$\text{vec3 } L = \text{lightPos.xyz} - P$~~

• texture (shadowMap, st)