

Gràfics 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

- glDrawElements

- Rasterització

- Transformació a Clip Space

glDrawElements

Transformació a 

Space Divisió de


perspectiva 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

ANTERIOR

(b) Fragment  Shader

POSTERIOR (c) dFdx, dFdy

POSTERIOR (d) Stencil Test

POSTERIOR

Exercici 3

El LOD 0 d'una textura té 1024 x 512 texels. Quina mida té el LOD 2 d'aquesta mateixa

textura?




LOD 0 → 1024 x 512 ; LOD 1 → 512 x 256; LOD 2 → **256 x 128**

Exercici 4

Tenim un cub representat amb una malla triangular formada per 8 vèrtexs i 12 triangles. Volem construir un VBO per representar aquest cub, 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 el cub aparegui il·luminat correctament).

(a) Quants vèrtexs necessitem representar al VBO?

$12 \text{ tri} * 3 \text{ v/tri} = 36$  **texs** (hem de repetir vèrtexs perquè no comparteixen normals)

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

$12 \text{ tri} * 3 \text{ index/tri} = 36$  **índexs**

Exercicis 5 i 6

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

M = modelMatrix

M^{-1} = modelMatrixInverse

= viewingMatrix

V^{-1} = viewingMatrixInverse

P = projectionMatrix

P^{-1} =

N = normalMatrix

projectionMatrixInverse

I = Identitat

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

 **P**

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

 **V^{-1}**

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

 **$V^{-1} * P^{-1}$**

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

 **$P * V$**

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

 **I**

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

 **M**

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

 **$V * M$**

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

 **N**

Exercici 7

Tenim activat alpha blending amb la

crida `glBlendFunc(GL_ONE_MINUS_SRC_ALPHA, GL_SRC_ALPHA)`

Hem produït un fragment amb color RGBA = (1.0, 0.5, 0.0, 0.2) corresponent al pixel (i,j). El color RGBA del pixel (i,j) al buffer de color és (1.0, 1.0, 0.5, 0.0). Indica quin serà el color RGBA resultant del blending, amb les operacions que duen a aquest resultat.

$$(1-0.2) (1.0, 0.5, 0.0, 0.2) + 0.2 (1.0, 1.0, 0.5, 0.0) = \\ (0.8, 0.4, 0.0, 0.16) + (0.2, 0.2, 0.1, 0) = \mathbf{(1.0, 0.6, 0.1, 0.16)}$$

Exercicis 8 i 9

Una forma d'expressar l'equació general del rendering és la següent:

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

(a) Què creus que representa λ ?

Longitud d'ona de la llum

(b) Què representa Ω ?



La semiesfera centrada al punt \mathbf{x} i alineada amb la normal \mathbf{n} que representa totes les possibles direccions en les que pot arribar llum a \mathbf{x} .

(c) Indica quin nom té la funció f_r

És el BRDF.


(d) Què són els tres primers paràmetres de la funció f_r ?

Els paràmetres són: posició, direcció d'entrada, direcció de sortida.

Exercici 10

Completa el següent FS per tal que calculi correctament el terme de Phong de la il·luminació:

```
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 NdotL = max( 0.0,
    dot( N,L ) ); float RdotV =
    max( 0.0, dot( R,V ) ); float Idiff
    = NdotL;
    float Ispec = 0;
    if (NdotL>0) Ispec = ........pow( RdotV, matShininess );

    return
        matAmbient * lightAmbient +
        matDiffuse * lightDiffuse * Idiff
        + matSpecular * lightSpecular *
        Ispec;
}
```

Exercici 11

Indica, en la notació estudiada a classe, $L(D|S)^*E$, quins light paths són suportats

per:

(a) Raytracing clàssic

LDS^*E i LS^*E



(b) Two-pass raytracing

LS^*DS^*E (i LS^*E)

Exercicis 12 i 13

Volem generar amb RayTracing una imatge 256x256 d'una escena interior tancada. Els objectes de l'escena estan configurats de forma que la probabilitat de que qualsevol raig intersecti un mirall és de 0.5 (l'altre 0.5 correspon a un objecte difós).

a) Quants rajos primaris caldrà traçar?

$$256 \times 256 = 2^{16} \text{ rajos.}$$

b) Quants rajos reflectits caldrà traçar, en total, si admetem un únic nivell de recursivitat (per exemple LDSE)?

$$= 2^{15} = 32768$$



c) Quants rajos reflectits caldrà traçar, en total, si admetem dos nivells de recursivitat (per exemple LDSSE)?

$$2^{15} + 2^{14} = 49152$$

d) Quants rajos primaris caldrà traçar si volem antialiasing amb 4 mostres per píxel?

$$4 \times 256 \times 256 = 2^{18} \text{ rajos.}$$

Exercici 14

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/m²)?

Irradiància



Exercici 15

Indica, per cadascuna de les següents magnituds, si afecta (SI) o no (NO) a la direcció del raig transmès, d'acord amb la Llei de Snell (considera també efectes indirectes):

(a) Velocitat de propagació de la llum als medis SI

(b) Longitud d'ona de la llum SI



(c) Angle d'incidència SI

(d) Color difós de la superfície (Kd) NO

Exercici 16

Tenim una aplicació que no suporta MipMapping, però volem simular el resultat de GL_LINEAR_MIPMAP_LINEAR en GLSL. Completa aquest codi, on lambda és un float amb el nivell de LOD (no necessàriament enter) més adient pel fragment:

```
vec4 sampleTexture(sampler2D sampler, vec2 texCoord, float lambda)
```

```
{  
    vec4 color0 = textureLod(sampler, texCoord, floor(lambda));  
    vec4 color1 = textureLod(sampler, texCoord, floor(lambda)  
+1);  
    return mix(color0, color1, fract(lambda));  
}
```

Podeu assumir que textureLod(P,sampler,lod) fa un accés bilineal a textura al punt P usant el nivell especificat a lod.

Exercici 17

A classe hem estudiat un algorisme per simular reflexions especulars en miralls plans basat en objectes virtuals. Explica clarament per què és necessari, en general, fer servir la versió amb stencil buffer.



Per tal que els objectes virtuals (reflectits) apareguin només a la porció del pla ocupada pel mirall.

Exercici 18

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



Completa la línia que necessitem al VS:

```
vtexCoord = 4.0 * texCoord;
```

Exercici 19

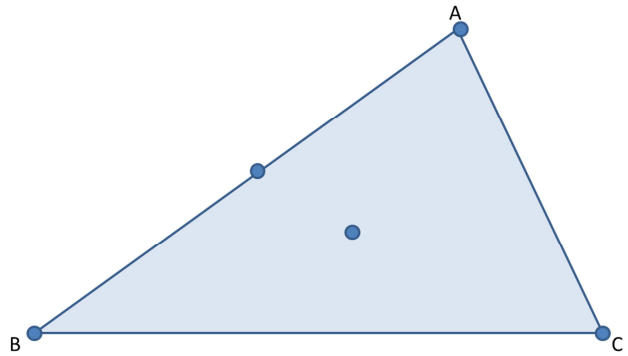
Indica les coordenades baricèntriques (α, β, γ) associades als vèrtexs A, B, C del triangle, pels punts que

(a) Baricent

$(1/3, 1/3, 1/3)$

(b) Punt mig de l'aresta AB

$(0.5, 0.5, 0)$



(c) Si un punt P té coordenades (α, β, γ) amb $\alpha = 0.2$ i $\beta = 0.3$, què podem dir de γ ?

$\gamma = 0.5$ (han de sumar 1)

(d) Si un punt P té coordenades (α, β, γ) amb $\alpha < 0$, què podem dir de P en relació al triangle?

P és exterior al triangle

Exercici 20

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

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

void main()
{
    vec3 L = normalize(lightPos - P);
    float NdotL = max(0.0, dot(N,L));
    vec4 color = vec4(NdotL);

    vec2 st = vtxCoord.st / vtxCoord.q;

    float storedDepth = texture(shadowMap, st).r;

    float trueDepth = vtxCoord.p / vtxCoord.q;

    if (trueDepth <= storedDepth) fragColor =
    color; else fragColor = vec4(0);
}
```