Examen parcial d'IDI 2023-2024, Q1. 27/10/2023 15:30

Nom i cognoms: Temps total: 1h 40'

Normativa preguntes curtes

1. Responeu les següents preguntes en el mateix full de l'enunciat.
2. Cal que les respostes siguin **clares, precises i concises**.
3. No es poden usar apunts ni calculadores ni cap dispositiu electrònic.

Escena 1: Una escena està formada per un terra al pla Y=0 amb dimensions 4x4 alineat amb els eixos X i Z i centrat a l'origen, un Patricio d'alçada 3 amb el centre de la base de la seva capsa mínima contenidora a la posició (-1,0,-1) mirant cap al eix +Z, i un Homer d'alçada 4 amb el centre de la base de la seva capsa mínima contenidora a la posició (1,0,1) mirant cap al eix -Z. El Patricio i el Homer estan escalats uniformement.

1. *(1 punt)* Tenint en compte que volem pintar l'**Escena 1**, resol els següents apartats amb el pseudo-codi adient:

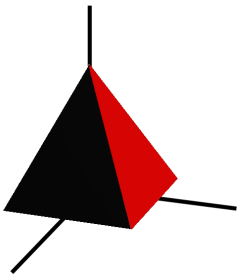
   a) Sabem que els punts mínim i màxim de la capsa contenidora del model del Homer són: (Hminx, Hminy, Hminz) i (Hmaxx, Hmaxy, Hmaxz), i que el model mira inicialment en direcció Z+. Completa la funció següent que calcula i retorna la TG necessària per pintar el Homer de l'escena.

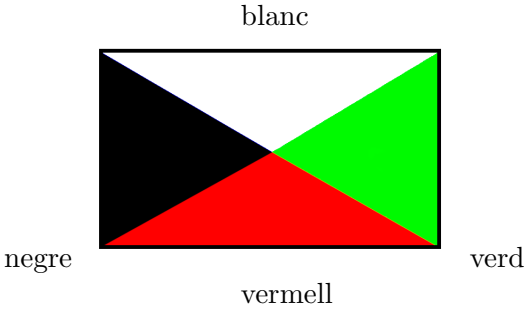
```
glm::mat4 TG_Homer()
{
    glm::mat4 TG = I;
    escala = 4/(Hmaxy-Hminy);
    cbase = ((Hminx+Hmaxx)/2, Hminy, (Hminz+Hmaxz)/2));
    TG = TG * translate (1,0,1);
    TG = TG * rotate_Y (180);
    TG = TG * scale (escala, escala, escala);
    TG = TG * translate (-cbase);
    return (TG);
}
```

   b) Suposant que el terra no necessita ser transformat, que també tenim la funció `TG_Patricio()` que ens retorna la TG necessària per pintar el Patricio de l'**Escena 1**, la funció `modelMatrix(TG)` que envia una TG al Vèrtex Shader, i que podem pintar cada model amb la funció `pintaModel(model)` on el model el podem indicar com `Homer, Patricio o Terra`, indica el pseudo-codi necessari per a que es pinti l'**Escena 1** al complet. Utilitza també la funció definida a l'apartat a).

```
TG1 = TG_Homer();
modelMatrix(TG1);
pintaModel(Homer);
TG2 = TG_Patricio();
modelMatrix(TG2);
pintaModel(Patricio);
TG3 = I;
modelMatrix(TG3);
pintaModel(Terra);
```

2. *(1 punt)* Tenim una escena amb una piràmide amb base quadrada de costat 2 i alçada 2, amb el centre de la seva base a l'origen de coordenades i el vèrtex (o l'àpex) sobre l'eix Y+. La cara orientada cap a X+ és vermella, la cara orientada cap a Z- és verda, la cara orientada cap a X- és blanca, la cara orientada cap a Z+ és negra, i la cara de la base orientada cap a Y- és blava. Completa els paràmetres d'una càmera en perspectiva que permeti veure en pantalla el rectangle de la dreta centrat al viewport i ocupant-lo sencer. Tenim un viewport (vista) de 800x400.






```
VM = lookAt (     (0, 4, 0)    , (0,0,0),     (-1, 0, 0)     );
PM = perspective (    2*atan(1/4)    ,   1   ,  2  ,  4  );
```

3. *(1 punt)* Si per visualitzar una escena posicionem una càmera amb OBS=(2,1,0), VRP=(0,1,0) i up=(0,0,1), indica el pseudo-codi necessari per calcular la mateixa View Matrix (VM) mitjançant angles d'Euler.

Solució:

```
VM = Translate (0,0,-2);
VM = VM * Rotate (-90, (0,0,1));
VM = VM * Rotate (-90, (0,1,0));
VM = VM * Translate (0,-1,0);
```

4. *(1 punt)* En una aplicació de diseny gràfic es genera un dibuix que té 5 colors utilitzats en la mateixa quantitat, codificats en RGB com: C1 = (1, 0, 0.5), C2 = (0, 0.8, 1), C3 = (0, 0.5, 0), C4 = (0, 1, 1) i C5 = (1, 0, 0).

   a) Tenim una impressora CMY que està carregada amb les tintes totes al màxim. Si s'imprimeix aquest dibuix repetidament, quina de les tintes de la impressora s'acabarà abans?
   Solució: La tinta que més s'utilitza és la cian (C) que és la que s'acabarà abans.

   b) Si a la impressora li falla la tinta cian (C) però continua imprimint el dibuix, de quins colors es veurà aquest dibuix? És a dir quins seran els valors, en RGB, dels colors C1, C2, C3, C4 i C5 un cop impressos en paper blanc?
   Solució:
   C1 = (1, 0, 0.5) i C5 = (1, 0, 0) ⇒ es veuran igual.
   C2 = (0, 0.8, 1) ⇒ es veurà = **(1, 0.8, 1)**.
   C3 = (0, 0.5, 0) ⇒ es veurà = **(1, 0.5, 0)**.
   C4 = (0, 1, 1) ⇒ es veurà = **(1, 1, 1)**.

   c) Suposant que la impressora anterior funciona correctament i té totes les tintes, indica de quin color es veuran les parts del dibuix dels colors C4 i C5 si aquest s'imprimeix usant paper groc de màxima intensitat.
   Solució: C4 es veurà de color verd = (0, 1, 0) i C5 es veurà igual perquè no varia.

   d) Tenint en compte els colors inicials del dibuix, si pensem en la seva representació en format HSB, quins d'ells tenen la saturació màxima (S = 1)?
   Solució: Tots

Examen parcial d'IDI 2023-2024, Q1. (Perm: A) 27/10/2023 15:30

Nom i cognoms:

Normativa del test

(a) A les graelles que hi ha a continuació, marca amb una creu les teves respostes de l'examen. **No es tindrà en compte cap resposta fora d'aquestes graelles**.

(b) No es poden usar apunts, calculadores ni cap dispositiu electrònic.

(c) Totes les preguntes tenen una única resposta correcta.

(d) Les preguntes contestades de forma errònia tenen una **penalització del 33%** del valor de la pregunta.

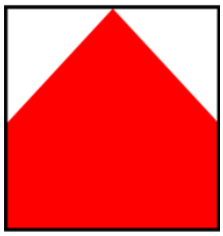
Num	A	B	C	D
5				
6				
7				
8				

Num	A	B	C	D
9				
10				
11				
12				

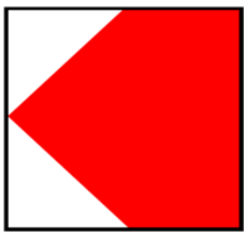
Num	A	B	C	D
13				
14				
15				
16				

5. *(0.5 punts)* Tenim una escena amb un triangle vermell amb vèrtexs V1 = (-2,-1,0), V2 = (2,-1,0) i V3 = (0,1,0). Suposant que hem inicialitzat la view matrix amb OBS = (0,0,0), VRP = (0,0,-2) i UP = (1,0,0), i la project matrix a la matriu identitat, indica quina de les següents imatges és la que sortirà en un viewport de 600x600 (sabem que el Vertex Shader i el Fragment Shader estan correctament implementats):

a)



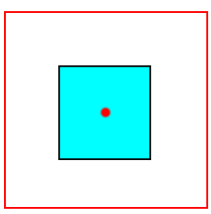
b)



c)

d)

6. *(0.5 punts)* Tenim un paral·lelepíped orientat amb els eixos coordenats de color blau, que té com a vèrtex mínim VMin = (-1,-1,0) i com a vèrtex màxim VMax = (1,1,4). Quins han de ser els paràmetres d'òptica d'una càmera ortogonal posicionada amb OBS = (2,0,2), VRP = (0,0,2) i up = (0,1,0) per a que la imatge que es vegi en un viewport quadrat de 600x600 sigui la que es veu a la imatge següent. Suposa que el quadrat que es veu ocupa la meitat del viewport.



a) left = -4; right = 4; bottom = -2; top = 2; Znear = 1; Zfar = 3;

b) left = 4; right = 0; bottom = -1; top = 1; Znear = 1; Zfar = 3;

c) left = -2; right = 2; bottom = -2; top = 2; Znear = 0.5; Zfar = 4;

d) left = -2; right = 2; bottom = -4; top = 4; Znear = 0.5; Zfar = 4;

7. *(0.5 punts)* Tenint en compte l'**Escena 1**, quina de les següents és la millor càmera per a que sigui una càmera en tercera persona que visualitzi tota l'escena, centrada i sense deformacions en un viewport amb ra_v=2.

   a) OBS = (0,10,0); VRP = (0,2,0); UP = (1,0,0); FOV = 60º; ra = 1; Znear = 4; Zfar = 8;
   b) OBS = (8,2,0); VRP = (0,2,0); UP = (0,1,0); FOV = 60º; ra = 1; Znear = 1; Zfar = 15;
   c) OBS = (0,2,8); VRP = (0,2,0); UP = (0,1,0); FOV = 60º; ra = 2; Znear = 2; Zfar = 15;
   d) OBS = (0,0,8); VRP = (0,0,0); UP = (0,1,0); FOV = 60º; ra = 2; Znear = 4; Zfar = 8;

8. *(0.5 punts)* Tenim una escena que es composa d'un únic objecte de 100 milions de triangles. Si assumim que aquest objecte és estàtic, és a dir que tindrà una única transformació geomètrica TG fixa durant tota l'execució de la nostra aplicació, quan ens convindrà més aplicar aquesta TG?

   a) Cada cop que pintem i just abans de pintar l'objecte.
   b) En el pipeline de visualització, dins del Vertex Shader.
   c) Si la TG és fixa i no varia no cal aplicar-la.
   d) Quan llegim el model de l'objecte i estem omplint la nostra estructura de dades del model.

9. *(0.5 punts)* Si tenim un model format per NC triangles i NV vèrtexs, quina diferència de memòria hi haurà entre guardar la seva topologia de forma explícita o fer-ho de forma implícita?

   a) Amb topologia implícita necessitarem guardar (9*NC) valors mentre que amb topologia explícita necessitarem (5*NC) valors.
   b) Amb topologia implícita necessitarem guardar (9*NC) valors mentre que amb topologia explícita necessitarem (3*NC + 3*NV) valors.
   c) Amb topologia implícita estalviarem més memòria perquè no guardem informació específica de les cares i aquesta es calcula al vol.
   d) Amb topologia explícita malgastem memòria perquè tenim informació repetida dels vèrtexs.

10. *(0.5 punts)* Tenim un Vertex Shader que té una variable de sortida `vec3 fcolor` al qual se li assigna el valor d'una variable `vec3 color` corresponent a les dades de color per vèrtex. El Fragment Shader recull `fcolor` com a variable d'entrada i l'assigna com a color de sortida fent `fragColor=vec4(fcolor, 1.0);`. El que veiem en el viewport és un triangle tot sencer de color blanc. Quins colors tenien els vèrtexs del triangle definits en el corresponent VBO?

    a) colorV1 = (1,0,0); colorV2 = (0,1,0); colorV3 = (0,0,1)
    b) colorV1 = (1,1,0); colorV2 = (0,1,0); colorV3 = (0,1,1)
    c) colorV1 = (0.25,0.5,0.25); colorV2 = (0.5,0.25,0.25); colorV3 = (0.25,0.25,0.5)
    d) colorV1 = (1,1,1); colorV2 = (1,1,1); colorV3 = (1,1,1)

11. *(0.5 punts)* Quina diferència pot haver entre la finestra OpenGL i el viewport?

    a) Poden tenir una relació d'aspecte diferent.
    b) Poden tenir una resolució diferent.
    c) Les dues respostes anteriors.
    d) Cap, ja que representen el mateix concepte.

12. *(0.5 punts)* Què fa la rasterització?

    a) Projecta en el viewport els vèrtexs de cada triangle d'acord amb la primitiva de pintat escollida.
    b) Genera els fragments que formen cada vèrtex.
    c) Genera els fragments que formen cada primitiva de pintat executada.
    d) Projecta i genera els píxels que formen cada triangle d'acord amb la primitiva de pintat escollida.

13. *(0.5 punts)* En relació al Clipping podem dir que:

    a) Descarta primitives que estan fora del volum de visió i pot generar o no noves primitives, retallant, si cal, les que es pinten dins.
    b) Transforma cada vèrtex del sistema de coordenades de observador al sistema de coordenades de clipping.
    c) Per cada triangle decideix si el seus fragments es pintaran o no perquè poden estar ocults per altres fragments.
    d) Per a cada vèrtex en coordenades de clipping comprova que no tingui cap valor negatiu.

14. *(0.5 punts)* Si no volem tenir deformació al mostrar una imatge resultant del procés de visualització. Què podem fer?

    a) Modificar adientment el window.
    b) Assegurar-nos que tant el window com el viewport tenen la mateixa relació d'aspecte.
    c) Modificar adientment el viewport.
    d) Qualsevol de les altres.

15. *(0.5 punts)* Per a què ens poden servir els estudis i gràfics que ens indiquen a quines zones de la pantalla del mòbil i amb quin grau de facilitat es pot arribar amb el polze?

    a) Per ajudar-nos a distribuir tots els nostres widgets únicament allà on es pot arribar fàcilment.
    b) Per col·locar millor el detector d'empremtes digitals del hardware.
    c) Per ajudar-nos a decidir on és més adient col·locar widgets que no volem que es premin per accident.
    d) Per ensenyar millor als usuaris com han d'agafar el mòbil.

16. *(0.5 punts)* Si una aplicació és molt complicada de fer servir pot passar que

    a) Sigui per culpa de la seva estètica agradable.
    b) Es produeixin menys errors al fer-la servir que amb una més senzilla.
    c) Gairebé ningú adquireixi el nostre producte.
    d) Cap de les altres és correcta.