Examen parcial d'IDI 2021-2022, Q2. 31/3/2022 10:30

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.

1. *(1 punt)* Tenim un dibuix amb els colors, codificats en RGB, C1 = (0.7, 0.7, 0), C2 = (1, 0.5, 1) i C3 = (0, 0.8, 0).

   a) Indica la codificació en CMY dels tres colors C1, C2 i C3
   Solució: C1 = (0.3, 0.3, 1), C2 = (0, 0.5, 0), C3 = (1, 0.2, 1)

   b) Si imprimim el dibuix amb una impressora CMY a la que li falla la tinta groga (i segueix imprimint), de quins colors es veurà el dibuix? Indica la seva codificació RGB.
   Solució: C1 = (0.7, 0.7, 1), C2 = (1, 0.5, 1), C3 = (0, 0.8, 1)

   c) En la codificació HSV, quin és el valor de la saturació (S) dels colors C1 i C3?
   Solució: S = 1 en tots dos casos.

   d) Si mirem el dibuix a través d'un filtre de color cian de quin color veurem el color C1? Indica la seva codificació RGB.
   Solució: el veurem de color verd (0, 0,7, 0)

2. *(1 punt)* Tenim una piràmide de base quadrada de costat 4 amb la base centrada al punt (0, 0, 0) i alçada de la piràmide 4 amb l'eix en direcció X+. Definim una càmera ortogonal amb paràmetres: OBS = (2, 0, 6), VRP = (2, 0, 0) up = (1, 0, 0), left = -2, right = 2, bottom = -2, top = 2.

   a) Quins són els valors de ZNear i ZFar necessaris per a què la piràmide quedi completament dins del volum de visió i de manera ajustada (volum mínim)?
   Solució: ZNear = 4; ZFar = 8.

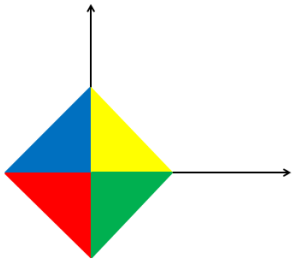
   b) Si tenim un viewport de 600x600 (amplada x alçada), quines coordenades té el vèrtex del pic de la piràmide (punt P) en els següents sistemes de coordenades?

```
P(SCO) =                    (Sist. Coord. Observador)
P(SCN) =                    (Sist. Coord. Normalitzades)
P(SCD) =                    (Sist. Coord. Dispositiu - només valors (x,y))
```

   Solució: Donat que el punt P = (4, 0, 0) en SCA, les dade demanades són:
   P(SCO) = (0, 2,-6); P(SCN) = (0, 1, 0); P(SCD) = (300, 600).

3. *(1 punt)* Tenim una escena formada per 4 triangles (veure imatge de l'esquerra): El primer triangle té vèrtexs v1.1=(0,0,0), v1.2=(4,0,0), v1.3=(0,4,0) i és de color groc. El segon té vèrtexs v2.1=(0,0,0), v2.2=(0,4,0), v2.3=(-4,0,0) i és de color blau. El tercer té vèrtexs v3.1=(0,0,0), v3.2=(-4,0,0), v3.3=(0,-4,0) i és de color vermell. I el quart té vèrtexs v4.1=(0,0,0), v4.2=(0,-4,0), v4.3=(4,0,0) i és de color verd.

   Completa els paràmetres necessaris d'una càmera ortogonal per a què al viewport, que és de 800 x 400, es vegi el que teniu a la imatge de la dreta (on el triangle que es veu és de color verd).



```
VM = lookAt (               ,  (4, -2, 0)  ,               );

PM = ortho (       ,       ,       ,       ,  1,  3);
```

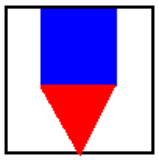
Solució:

```
VM = lookAt ( (4, -2, 2) , (4, -2, 0) , (0, -1, 0) );
PM = ortho (-4 , 4 , -2 , 2 , 1, 3);
```

4. *(1 punt)* Tenim una piràmide de base quadrada de costat 2, amb la base centrada al punt (0,0,-1) i alçada de la piràmide 2 amb l'eix en direcció Z-. A l'escena tenim també un cub de costat 2 centrat a l'origen. El viewport esta definit amb glViewport (0,0,600,600). Es fa la següent inicialització d'una càmera ortogonal (posició + orientació i òptica):

```
PM=ortho(-2, 2, -2, 2, 2, 4);
projectionMatrix (PM)
VM=translate (0, 0, -3);
VM=VM*rotate (90, 0, 0, 1);
VM=VM*rotate (90, 0, 1, 0);
VM=VM*translate (0, 0, 1);
viewMatrix (VM);
```

Solució:



Dibuixa el viewport i el que s'hi veurà dins.

Examen parcial d'IDI 2021-2022, Q2. (Perm: A) 31/3/2022 10:30

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				

Descripció d'una escena que servirà per a alguns exercicis del test

Escena 1: Tenim una escena formada per: un terra modelat per un quadrat de costat 10 ubicat en el pla Y=0, centrat a l'origen i de costats paral·lels als eixos X i Z; un Patricio (Pat1) d'alçada 3 amb el centre de la base de la seva capsa contenidora al punt (-3, 0, 3) i mirant cap a Z+; i un segon Patricio (Pat2) d'alçada 6 amb el centre de la base de la seva capsa al punt (0, 0, 0) i mirant en direcció (1, 0, 1). Els dos Patricios estan escalats uniformement.

5. *(0.5 punts)* Els editors d'imatges sempre tenen una opció que permet modificar qualsevol imatge passant-la a **escala de grisos**. A nivell dels models de color, això ho podem aconseguir...

   a) Afegint color gris (R, G i B en la mateixa quantitat) a tots els colors de la imatge
   b) Reduïnt la saturació (component S de HSV) a 0
   c) Pintant tot del mateix color gris RGB = (0.5, 0.5, 0.5)
   d) Cap de les altres respostes és correcta

6. *(0.5 punts)* Indica quin dels següents grups de paràmetres d'una càmera perspectiva permet veure l'**Escena 1** centrada, sense retallar ni deformar i de manera que la seva esfera contenidora ocupi el màxim del viewport (càmera en tercera persona). El Patricio Pat1 s'ha de veure de cara amb una direcció de visió paral·lela a l'eix Z de l'aplicació. El viewport és de 600 x 600.

   a) RadiEscena = sqrt(59); OBS = (0, 3, 2 * RadiEscena); VRP = (0, 3, 0);
   FOV = atan(1.0/2.0); ra = 1; ZNear = RadiEscena.
   b) RadiEscena = sqrt(236)/2; OBS = (8, 3, 8); VRP = (0, 0, 0);
   FOV = atan(1.0/2.0); ra = 2; ZNear = 3 * RadiEscena.
   c) RadiEscena = sqrt(236)/2; OBS = (8, 3, 8); VRP = (0, 0, 0);
   FOV = asin(1.0/2.0); ra = 2; ZNear = 3 * RadiEscena.
   d) RadiEscena = sqrt(59); OBS = (0, 3, 2 * RadiEscena); VRP = (0, 3, 0);
   FOV = asin(1.0/2.0); ra = 1; ZNear = RadiEscena.

7. *(0.5 punts)* En el procés de visualització projectiu, definits un vertex i fragment shader, quina de les següents respostes mostra l'ordre correcte en el que s'executen certes accions en enviar una primitiva a pintar?

   a) Pas a coordenades de clipping, pas a coordenades de dispositiu, rasterització, z-buffer.

   b) Pas a coordenades de dispositiu, retallat (clipping), rasterització, z-buffer.

   c) Pas a coordenades de clipping, rasterització, retallat (clipping), càlcul color per a cada fragment.

   d) Pas a coordenades d'observador, z-buffer, retallat (clipping), càlcul color per a cada fragment.

8. *(0.5 punts)* Tenim una càmera en tercera persona correctament inicialitzada per visualitzar l'**Escena 1**. Afegim a l'escena un arbre de 10 unitats d'alçada situat amb la seva base sobre el terra. Indica quin dels següents grups de paràmetres ens servirien per modificar la càmera de manera que continui veient-se centrada, sencera, sense deformar i ocupant el màxim del viewport.

   a) FOV

   b) FOV, OBS

   c) FOV, OBS, VRP

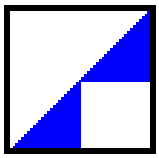
   d) OBS, VRP, ra

9. *(0.5 punts)* Dibuixem un triangle amb vèrtexs V1 = (-1, -1, 0), V2 = (1, -1, 0) i V3 = (1, 1, 0). El viewport és de 600 x 600 i els shaders amb els que pintem el triangle tenen el següent codi:

```
Vertex Shader:

in vec3 vertex;

void main() {
   gl_Position = vec4(vertex,1);
}
```

```
Fragment Shader:

out vec4 FragColor;

void main () {
   // codi per donar color al fragment
}
```

   Si el resultat del pintat d'aquest triangle és el que teniu a la imatge següent, quin és el codi que li falta al Fragment Shader?



   a)
```
if (gl_FragCoord.x > 300 && gl_FragCoord.y < 300) discard;
FragColor = vec4 (0, 0, 1, 1);
```

   b)
```
if (gl_FragCoord.x < 300 && gl_FragCoord.y > 300) discard;
else FragColor = vec4 (0, 0, 1, 1);
```

   c)
```
if (gl_FragCoord.x > 300 && gl_FragCoord.y > 300) discard;
FragColor = vec4 (0, 0, 1, 1);
```

   d) Cap dels codis de les altres respostes serveix

10. *(0.5 punts)* Volem pintar l'**Escena 1**. Tant el model del terra com el del Patricio tenen assignat color per vèrtex. Quants VAOs i VBOs necessitem per pintar l'escena?

    a) 2 VAOs i 2 VBOs

    b) 2 VAOs i 4 VBOs

    c) 3 VAOs i 3 VBOs

    d) 3 VAOs i 6 VBOs

11. *(0.5 punts)* Tenim un model d'un cub de costat 1 centrat a l'origen de coordenades. Volem pintar dos cubs, un de costat 2 centrat a l'origen, i un altre de costat 1 centrat al punt (1.5, 0, 0), Quin dels següents codis pintarà els cubs d'aquesta manera?

a)
```
TG1 = scale(2,2,2);
TG2 = translate(1.5, 0, 0);
modelMatrix(TG1);
pintaCubo();
modelMatrix(TG2);
pintaCubo();
```

b)
```
TG1 = scale(2,2,2);
modelMatrix(TG1);
TG2 = translate(1.5, 0, 0);
modelMatrix(TG2);
pintaCubo();
pintaCubo();
```

c)
```
TG1 = scale(2,2,2);
modelMatrix(TG1);
pintaCubo();
TG2 = TG1 * translate(1.5, 0, 0);
modelMatrix(TG2);
pintaCubo();
```

d) Cap de les altres és correcta

12. *(0.5 punts)* Volem definir una càmera que permeti veure l'**Escena 1** centrada en el viewport i amb el Patricio **Pat2** mirant de front a la càmera. Quin dels següents trossos de codi ens permet definir la view matrix (VM)?

a)
```
glm::vec3 OBS (20, 0, 20);
glm::vec3 VRP (0, 0, 0);
glm::vec3 up (0, 1, 0);
VM = lookAt (OBS, VRP, up);
```

b)
```
glm::vec3 OBS (-20, 3, -20);
glm::vec3 VRP (0, 3, 0);
glm::vec3 up (0, 1, 0);
VM = lookAt (OBS, VRP, up);
```

c)
```
VM = Translate (0, 0, -20);
VM = VM * Rotate (-45, 0, 1, 0);
VM = VM * Translate (0, -3, 0);
```

d)
```
VM = Translate (0, 0, -20);
VM = VM * Rotate (45, 1, 0, 0);
VM = VM * Rotate (45, 0, 1, 0);
VM = VM * Translate (0, -3, 0);
```

13. *(0.5 punts)* En l'àmbit de la usabilitat, l'autonomia està relacionada amb:

a) no molestar a l'usuari amb missatges d'error massa llargs

b) utilitzar contrastos adients entre el text i el fons

c) informar a l'usuari sobre l'estat de les tasques en tot moment

d) anticipar-se a les seves necessitats

14. *(0.5 punts)* Suposem que volem afegir un Legoman a l'**Escena 1**. Aquest Legoman ha d'estar situat al damunt del Patricio Pat2, de manera que el centre de la base de la capsa contenidora del Legoman coincideix amb el centre de la cara del damunt de la capsa contenidora del Patricio Pat2. El Legoman també ha d'estar mirant en la mateixa direcció que Pat2 i ha de fer alçada 3 (escalat uniformement). Sabent que inicialment el Legoman està mirant cap a Z+ i que la capsa contenidora del model té punt mínim (Lxmin, Lymin, Lzmin) i màxim (Lxmax, Lymax, Lzmax), quin dels següents trossos de codi calcula la TG que cal aplicar-li al model del Legoman per a què aparegui a l'escena com volem?

a)
```
TG = translate (0, 6, 0);
TG = TG * Rotate (45, 0, 1, 0);
TG = TG * Scale (3/(Lymax-Lymin), 3/(Lymax-Lymin), 3/(Lymax-Lymin));
TG = translate (-(Lxmin+Lxmax)/2, -(Lymin+Lymax)/2, -(Lzmin+Lzmax)/2);
```

b)
```
TG = translate (-(Lxmin+Lxmax)/2, -(Lymin+Lymax)/2, -(Lzmin+Lzmax)/2);
TG = TG * Scale (3/(Lymax-Lymin), 3/(Lymax-Lymin), 3/(Lymax-Lymin));
TG = TG * Rotate (45, 0, 1, 0);
TG = translate (0, 7.5, 0);
```

c)
```
TG = translate (0, 6, 0);
TG = TG * Rotate (45, 0, 1, 0);
TG = TG * Scale ((Lymax-Lymin)/3, (Lymax-Lymin)/3, (Lymax-Lymin)/3);
TG = translate (-(Lxmax-Lxmin)/2, -Lymin, -(Lzmax-Lzmin)/2);
```

d)
```
TG = translate (0, 7.5, 0);
TG = TG * Rotate (45, 0, 1, 0);
TG = TG * Scale (3/(Lymax-Lymin), 3/(Lymax-Lymin), 3/(Lymax-Lymin));
TG = translate (-(Lxmin+Lxmax)/2, -(Lymin+Lymax)/2, -(Lzmin+Lzmax)/2);
```

15. *(0.5 punts)* Suposem una càmera amb paràmetres OBS = (10, 3, 0), VRP = (0, 3, 0) i up = (0, 1, 0) per veure l'**Escena 1**. Quins dels següents paràmetres d'òptica permetran veure l'ecena sense retallar ni deformar en un viewport de 800 x 400?

a) FOV = 60, ra = 1.0, ZNear = 5, ZFar = 10

b) FOV = 90, ra = 0.5, ZNear = 5, ZFar = 15

c) left = -6, right = 6, bottom = -3, top = 3, ZNear = 5, ZFar = 15

d) left = -3, right = 3, bottom = -6, top = 6, ZNear = 3, ZFar = 20

16. *(0.5 punts)* La inconsistència induïda s'utilitza per:

a) mostrar a l'usuari que hem modificat el funcionament d'algun element d'una aplicació

b) diferenciar de forma evident entre botons d'una aplicació que tenen funcionalitats oposades

c) modificar el *look & feel* entre plataformes per facilitar l'aprenentatge

d) evitar l'ús d'estructures invisibles en aplicacions de mòbil